A social-technical approach to selecting software supporting COTS-Based Systems

Douglas Kunda

Submitted for the Degree of Doctor of Philosophy

University of York
Department of Computer Science

October 2001

Abstract

Developing countries (DCs) have yet to fully benefit from the many advances in the Information Technology (IT) field, mainly due to problems such as lack of resources and systems infrastructure. COTS-Based Systems (CBS) entail purchasing a number of commercial off-the-shelf (COTS) software components, each satisfying some part of the requirements of the system and integrating these components into the required system. CBS offers a number of benefits that the DCs can access, such as reducing development and maintenance costs. Therefore, CBS can be used to provide support for DCs to improve their IT processes.

The motivation for this thesis is an interest in improving the process associated with software systems development and procurements for organisations in DCs using CBS approaches. The research work focussed on investigating what processes (including traditional and soft factors) provide support for evaluating and selecting software components for CBS. This was achieved by three studies supported by a literature review.

The first study was aimed at eliciting and synthesising current CBS practices from the UK and Zambia, which brought out problems associated with CBS and resulted in a focussed research direction. The second study identified and classified important processes (including traditional and soft factors) that support COTS software evaluation and selection for CBS from the UK, which facilitated the development of a social technical framework for COTS software evaluation (STACE). In the third study the framework was evaluated in Zambia. The findings of the third study indicate that, though the framework had some limitations, it was found useful, useable, satisfied user needs and valid for use.

The main contribution of this research is STACE, a generic social-technical framework for COTS software evaluation and selection. It facilitates the examination of relationships between factors in different processes and their impact on COTS software evaluation and selection success, and therefore it can be used for further academic research. Furthermore, the framework can be used confidently to plan and implement COTS software evaluation and selection for CBS.

Table of Contents

Table of Figures

Table of Tables

Acknowledgements

First, I would like to thank my supervisor Laurence Brooks for providing excellent academic support and for his sincere friendship throughout my DPhil research. I also wish to express my most sincere thanks to everyone who contributed or participated in the empirical research. Many thanks are due to my employers, the Environmental Council of Zambia and the Director, Mr James S. Phiri for their moral and financial support. I also thank the Norwegian Embassy in Zambia for their financial support.

I would like to acknowledge the invaluable support of members of the Management Information Systems (MIS) group that kept me motivated during the difficult times. I would like to thank my special friends from Church who were a source of inspiration and in particular Pastor Richard Brooks (York), Pastor Conrad Mbewe (Lusaka) and Bestone Chileya (Lusaka).

Finally, special thanks are due to my wife Phyllis for her love, encouragement and support during this most difficult time of my career. Also special thanks go to my daughters Mwelwa and Seda, for their love and understanding especially when I had to spend long hours in my study. Above all, I thank God for sustaining and helping me throughout my time in the UK. The thesis is,

dedicated to the memory of Mr John Kunda Maliseni

my late father, who unexpectedly passed way in June 1999.

Declaration

The author carried out the research project described in this thesis during the period, October 1997 to May 2001.

The results presented in Chapters 5, 6 and 7 has been presented at a number of conferences and published in journals. A list of publications in which the material has subsequently appeared is attached as Appendix 1.

1 Introduction

This chapter provides an introduction to the field of research presented in this thesis. It contains the motivation for selecting this research area and the importance of this research topic. The central research question of the thesis is presented along with the objectives that arise from the central research question. The research strategy and the expected contribution of the research are described. Finally the chapter concludes with an overview of the remaining chapters of the thesis.

1.1 Background

The diffusion of computer technology in developing countries (DCs) is at an embryonic stage and does not impinge on the overwhelming majority of the people (Corr, 1995). A range of factors, including various infrastructural, financial, political and cultural aspects have acted against the effective development and exploitation of information technology. As a way to combat this organisations in DCs are turning to Commercial-Off-The Shelf (COTS) software components because modern information systems are becoming increasingly expensive to build and maintain.

Building systems from COTS software components can potentially be used to reduce software development and maintenance costs, as well as reducing software development time by bringing the system to market as early as possible (Clements, 1996; Haines *et al*, 1997). Therefore, it has a higher potential to benefit DCs compared with other systems development approaches. Although opinion vary on a precise definition for COTS, COTS software (also known as generic products, shrink-wrapped and commercial software) can be seen as referring to all software sold as tradable products (purchased from a vendor, distributor or store) for all computer platforms including mainframes, workstations and microcomputers (Sawyer, 2000).

1.2 Outline of research field

This thesis focuses in the field of social-technical approaches to information systems, information systems in DCs and COTS-Based Systems (CBS) development. An outline of each research field is provided in this section.

1.2.1 Social-technical approaches to information systems

Information systems (IS) are the means by which people and organisations, utilising technologies, gather, process, store, use and disseminate information systems (UKAIS, 1999). IS can be used to help an organisation to achieve improved efficiency of its operations and effectiveness through better managerial decisions (Avison and Fitzgerald, 1995). It can be used by organisation to support its business operations, support of managerial decision making and support of strategic competitive advantage (O'brien, 1999)

Despite attempts to make software development more rigorous, a considerable proportion of computer system development effort results in products that do not provide user satisfaction (Vidgen, 1997). Examples include, the CONFIRM reservation system (Oz, 1994), Australian government IS project - Mandata (Sauer, 1993) and the London ambulance service computer-aided despatch system project (Beynon-Davies, 1999). A number of social and organisational issues have been identified to affect software systems success, such as user resistance (Lyytinen and Hirschheim, 1987), group interaction problems (Bjorn-Andersen, 1988) and organisational power and politics (Avgerou and Cornford, 1998).

The social-technical approach is said to be an important strategy for addressing organisational and social issues (Mumford, 1990). For example, many authors argue that resistance to the system can be addressed through user participation in the software process because this can give a sense of user ownership and acceptance (Gronbaek *et al*, 1993; Axtell *et al*, 1997). The social-technical approach is aimed at developing a system that consists of both the human subsystem and technical subsystem in an integrated manner, such that the integrated system functions in an optimal way (Wieringa, 1996).

1.2.2 Information systems in developing countries

A number of models and approaches for informations systems implementation have been adopted for DCs, for example a methodology based on modification of Multiview (Bell, 1996). However, organisations in DCs have not fully realised the potential benefits of IS because problems unique to them. For example, in Zambia there is a critical shortage of skilled human resources in IT, scarcity of foreign currency to import hardware and software, and poor infrastructure (Shitima, 1990; Jere, 1992; Corr, 1995). Furthermore, most systems and methods developed in the developed countries do not work satisfactorily in DCs because of different socio-cultural context and therefore require to be adjusted accordingly (Bjorn-Andersen, 1990; Janczewski, 1992).

A number of strategies have been developed to help organisations in DCs to address these problems and promote IT in DCs. The strategies include international aid (Odedra, 1995), technology transfer (Robey *et al*, 1990) and government policies (Heeks, 1999). For example, formulation of national IT policy can used by the government to set IT goals and direction for development and also as an instrument to monitor acquisition, usage, standards and human resource development and the development of local computing industry (Ojo, 1992; Jere, 1992; Corr, 1995). However, there are still problems with IS development in DCs because most of these strategies require financial resources which is not available in DCs.

1.2.3 COTS-based systems

COTS-Based Systems (CBS) development is a process of building software systems by integrating pre-existing multiple COTS software components each satisfying some part of the requirements of the system (Brown and Wallnau, 1996b; Vidger *et al*, 1996). Building systems from COTS software components offers the opportunity to lower costs by sharing them with other users and has potential for reduced training and infrastructure costs (Oberndorf, 1997; Braun, 1999). Therefore, by employing CBS, organisations will not spend too much time on developing expensive systems, with only one customer to bear the development and maintenance costs over the life of the system. Furthermore, CBS offers the capabilities of extending and tailoring COTS software products through APIs, plug-ins and scripting languages (Vigder and Dean, 1997). Therefore, CBS has great potential for application in DCs.

CBS can be partitioned into the following essential activities: requirements engineering, component evaluation and selection, component adaptation, component integration and system evolution (Brown and Wallnau, 1996b; Tran *et al*, 1997). Requirements engineering assists in establishing a basis for evaluating and selecting appropriate COTS software candidates. Component qualification ensures that a candidate component will perform the functionality required and will exhibit the quality characteristics (e.g. performance, reliability, usability) that are required. Component adaptation usually involves extending and tailoring the capabilities of the COTS software products through APIs, plug-ins and scripting languages. Component integration is the practice of assembling a set of software components/subsystems to produce a single, unified software that supports some need of an organisation.

1.3 Research Motivation

Developed nations have used IT to help them change the way they do business so as to give them a strategic advantage in their operations (e.g. the use of ATMs in banks to improve customer service). However, the investment returns in DCs have fallen short of their potential due to problems that are characteristic of DCs, such as low income per capita. Solving these problems will require significant resources and government policies to tackle the underlying causes of the problem. Therefore, the motivation for this research is not to solve these problems; rather it is to establish how CBS development can provide support for organisations in DCs to improve the processes associated with software systems development and procurements. This initial interest emerged because of the researcher's previous participation in development of information systems and COTS-based systems in DCs.

Building of systems from COTS software depends on successful evaluation and selection of COTS software to meet customer requirements (Maiden and Ncube, 1998). A number of problems associated with COTS software evaluation and selection have been identified in literature. For example, rapid changes in market place (Carney and Wallnau, 1998); lack of well-defined process (Kontio, 1996); "black box" nature of COTS components (Vigder *et al*, 1996); and misuse of data consolidation method (Morisio and Tsoukias, 1997; Maiden and Ncube, 1998). Therefore, the motivation for this research is an interest in understanding what important processes and factors provide support for COTS software selection. This would provide insight into causes of COTS software selection problems and contribute to reducing risk associated with CBS development.

1.4 Central research question

The ultimate objective of this research is to contribute towards reducing risks and costs associated with CBS development. This can be achieved by improving COTS software evaluation and selection processes. COTS selection is a process of determining "fitness for use" of previously developed components that are being applied in a new system context (Haines *et al*, 1997). The evaluation of software components can also extend to include qualification of the development process used to create and maintain it (for example, ensuring algorithms have been validated, and that rigorous code inspection has taken place) (Brown and Wallnau, 1996b).

A number of frameworks for COTS software evaluation and selection have been developed aimed at addressing problems associated with COTS software selection. Useful examples include software system evaluation framework (Boloix and Robillard, 1995); off-the-shelf option (Kontio, 1996); Delta technology framework (Brown and Wallnau, 1996a) and procurement-oriented requirements engineering (Maiden and Ncube, 1998).

However, what is missing in these frameworks is the "soft" issues or the non-technical issues such as costs, organisational issues, vendor capability and reputation (Powell *et al*, 1997). Oberndorf *et al* (1997) highlight the usefulness defining the criteria to include such issues as vendor's time in business, responsiveness to customers and willingness to support their product. Nevertheless, little effort has been directed towards identifying and classifying important processes and factors supporting COTS software selection for practical use. Therefore, the central research question can be framed as:

> ***What processes (including traditional and soft factors) provide support for evaluating and selecting software components for COTS-based systems?***

In order to answer this research question, three immediate objectives were formulated. First, to achieve a more comprehensive understanding of how CBS can provide support for organisations, not only by studying the potential benefits and risks associated with CBS, but also by eliciting current CBS practices. Second, to identify important processes and factors that supports COTS software component selection in CBS. Lastly, to provide a generic social-technical framework for COTS software evaluation and selection that supports CBS.

In this thesis, a process is defined as a collection of related tasks leading to a product, for example the requirements definition process comprises a number of tasks (and activities) resulting in requirements documents. Organisational practices are established procedures, methods and approaches adopted by organisations in achieving a particular task and represent in some way the lessons they have learned about how best to achieve that particular task. For example, some organisations use prototyping to elicit customer requirements while others use brainstorming meetings. Thus a process will comprise a number of related organisational practices. A factor is a circumstance or influence contributing to a result, for example cost is an important factor in COTS software selection. Therefore, a number of factors could be associated with organisational practices and a process.

1.5 Research strategy

The research process consists of four main stages 1) literature review, 2) first study to elicit current CBS practices, 3) second study to identify important processes and factors that support COTS software selection and to develop a theoretical framework, and 4) third study to evaluate the theoretical framework.

Literature review involved an analysis of literature on CBS, COTS software evaluation and selection, social-technical approaches, information systems in DCs and other relevant topics. The potential contribution of the social-technical approaches to information systems development was elicited during this process. The review of literature on information systems in DCs for example assisted in understanding of the context, current practices and problems that are unique to DCs. The problems associated with building systems from COTS software components were identified and theoretical background in software engineering was established. Literature review was also the basis for developing the field studies.

The first study involved a survey study in the UK and Zambia to elicit and synthesise current practices and potential benefits of CBS. The survey was founded on literature review activities. As a result of this study a more comprehensive understanding of the current situation, problems (and solutions) people have experienced in relation to CBS were elicited. For example, the problem of COTS software evaluation and selection was brought out in the first study. This resulted in a focussed research project and better ways to help organisations in Zambia to develop and implement information systems. The findings from this study formed the basis for adapting the framework for evaluating COTS software for the Zambian context.

The second study was aimed at identifying important processes (including traditional and soft factors) that support COTS software selection. A series of interviews was used to identify important processes/factors and eight organisations from the UK participated in the study. Explanation building was used to analyse the data and the identified factors were classified into processes. This facilitated the development of a social-technical approach to COTS software evaluation (STACE) framework. The STACE framework contains the following characteristics:

- Support for a *systematic approach* to COTS evaluation and selection. Most organisations select their COTS components in an ad-hoc manner (Kontio, 1996). There is need for example to reuse lessons learnt from previous evaluation cases by maintaining a database of evaluation results.
- Support for *evaluation of both COTS products and the underlying technology*. Most COTS evaluation frameworks emphasise either on COTS products evaluation or technology evaluation. This method proposes keystone evaluation strategy (Obarndorf, 1997) in which the underlying technology is selected before selecting the COTS products.
- Use of *social-technical techniques* to improve the COTS software selection process. This has been greatly influenced by the social-technical school and work by (Mumford, 1995). The STACE recommends the use of a social-technical evaluation criteria and customer participation in the COTS selection process.
- Use of *multi-criteria decision-making techniques* to consolidate evaluation attribute data. The STACE proposes the use of Analytic Hierarchy Process (AHP) which was developed by (Saaty, 1990) and successfully used in software selection

(Zviran, 1993; Kontio, 1996).

In the third study, the effectiveness of the STACE framework was evaluated in Zambia. A workbook that operationalises the STACE framework was developed, then tailored to the Zambian context and used to support the evaluation process. A multiple-case study strategy was used to evaluate the STACE framework. The outcome of this study led to the confirmation of the validity of the STACE framework for selecting COTS components supporting CBS. The general lessons from this evaluation exercise were extracted to establish good practice and learning for current and future research.

1.6 Research contribution

As explained previously, there is no existing framework for COTS software selection, which addresses the non-technical issues adequately. This study aim at filling this research gap by studying what important processes (traditional and soft factors) provides support to selection of software for CBS. The expected contribution of the study to the body of knowledge is two-fold.

First, the study provides a generic social-technical framework for COTS software evaluation and selection. The framework provides a classification of important processes (including traditional and soft factors) that support COTS software selection. It highlights relationships between processes (and factors within each process) and thus facilitates the examination of relationships between factors in different processes and their impact on success of COTS software selection. Therefore, it can be used for current and future academic research. In addition, the framework provides guidance for the process of selecting COTS software and incorporates the often-neglected non-technical issues such as vendor reputation. Therefore, it can be used to confidently plan and implement COTS software selection and contribute to reducing risks associated with CBS.

Second, the study provides a deeper understanding of how social-technical approaches address information system problems and the potential benefits of CBS for DCs. In particular, the study provided some valuable insights into COTS software evaluation and selection, and the applicability of this to developing countries.

1.7 Thesis organisation

This thesis comprises 8 chapters and 7 appendices. Chapters 2 and 3 are literature review chapters. Chapter 2 contains a review of the literature on social-technical approaches to information systems and information systems in DCs. The human, social and organisational issues that affect the success of software systems are described. It discusses common approaches to information systems, focussing on the social-technical approaches. Information systems in DCs and the problems of implementing information systems in DCs are described. The chapter also examines current approaches adopted by DCs to implement information systems and finally an example of a developing country is provided.

Chapter 3 reviews literature on CBS with focus on evaluation and selection of COTS software components to support CBS. The benefits and risks associated with CBS are described. The COTS software evaluation problems, process, method and techniques are discussed in this chapter. Finally, the most important frameworks for evaluating and selecting COTS components are described.

Chapter 4 describes the research methods used in different phases of the research. In addition, this chapter will discuss the research design, unit of analysis and provide details of the data collection and analysis procedures and techniques.

Chapter 5 presents the results of the first study to elicit and synthesise current CBS practices from the UK and Zambia. It provides an understanding of current practices, benefits and risks associated for building systems using COTS software. The chapter draws out both the similarities and differences between the UK and Zambia, and discusses their significance on the overall research.

Chapter 6 provides a detailed description of the second field study aimed at identifying important processes and factors that support COTS software selection. It provides an extensive list of processes and factors that support COTS software evaluation and selection. It also provides a summary of the social-technical framework for COTS software evaluation (STACE), which resulted from the second study.

Chapter 7 presents the results of the third study aimed at evaluating STACE framework in Zambia. It describes the modification of the framework to the Zambian context and the main findings of the evaluation exercise. The main purpose of the evaluation exercise is to assess the effectiveness of the framework to support COTS software selection.

The concluding chapter, Chapter 8 provides an overview of the whole study and a summary of the important research findings. A review of the STACE framework is also provided in this chapter. This chapter includes implications of the research findings for researchers and practitioners, particularly those involved in COTS software selection. The limitations of the study and future research directions are also provided in this chapter.

2 Information Systems and Developing Countries

The chapter introduces the field of information systems and discusses the human, social and organisational issues that affect the success of software systems. The chapter will also discuss common approaches to information systems, focussing on the social-technical approaches. Information systems in DCs and the problems of implementing information systems in DCs will be discussed. The chapter will also examine current strategies adopted by DCs to implement information systems and finally an example of a developing country is provided.

2.1 Introduction

Information systems (IS) can be used to help an organisation to achieve improved efficiency of its operations and effectiveness through better managerial decisions (Avison and Fitzgerald, 1995). It can help provide strategic products and services that give a business organisation a comparative advantage over its competitors (O'brien, 1999). Many organisations in DCs today realise that information systems offer potential benefits such as cost savings through increased efficiency of operations and effectiveness in running of their organisations (Grant-Lewis and Samoff, 1992).

However, software systems do not exist in isolation, they are used in social and organisational contexts (Sommerville, 1995). Experience and many studies show that the major cause of most software failures is the social and organisational factors rather than technical issues (Potts, 1993; Friedman and Kahn, 1994; Beynon-Davies, 1999). Social-technical approaches have been developed to deal with some of these human, social and organisation issues. Social-technical development is oriented to developing both social and technical subsystems in an integrated way, so that the integrated system functions in an optimal way (Wieringa, 1996). Examples of methods based on a social-technical development approach include Multiview (Avison and Wood-Harper, 1990) and ETHICS (Mumford, 1990).

Although IS have been successfully used to help organisations in developed countries to achieve improved efficiency, most DCs have not yet fully benefited because of problems experienced by them. Examples include lack of adequate skilled human resources, economic constraints, systems infrastructure deficiency, social cultural and applications problems (Bogod, 1979; Okot-uma, 1992; Corr, 1995). A number of strategies have been adopted by DCs to address these problems including dependence on international aid agencies, government policies and technology transfer (Odedra, 1995; Heeks, 1999; Hassan, 2000).

The next section will introduce the field of information systems and social technical approaches, to provide the context in which the research is conducted.

2.2 Information systems and social-technical approaches

This section define an information system as used in this study, discusses the social and organisational problems associated with IS and then introduces common approaches to IS, focussing on social-technical approaches.

2.2.1 What is an information system?

There are a number of definitions of information systems in literature. For example, Hicks (1993) define IS as a formalised computer information system that can collect, store, process, and report data from various sources to provide the information necessary for management decision making. Laudon and Laudon (1995) define IS as a set of interrelated components that collect (or retrieve), process, store and disseminate information to support decision making, control, analysis and visualisation in an organisation. The important idea (assumption) brought out by these definitions is that IS are the means by which information is provided (Flynn, 1998). Information has been defined as data arranged in a meaningful way for some perceived purpose and it implies relevance to a consumer (user) of information (Liebenau and Backhouse, 1990).

However, this thesis adopts the UKAIS (1999) definition of Information Systems:

> *Information systems are the means by which people and organisations, utilising technologies, gather, process, store, use and disseminate information systems. The domain involves the study of theories and*

practices related to the social and technological phenomena, which determine the development, use and effects of information systems in organisations and society.

This definition has been adopted because it incorporates the three important perspectives of IS namely the human dimension, organisations and technology. The human perspective highlights various needs of the individual that use information technology to perform the jobs, for example, the human-computer interaction (Avgerou and Cornford, 1998). Information systems are used in organisations, which are composed of different structures, goals, politics and unique culture and therefore the organisations perspective would effect the information systems (Laudon and Laudon, 1995; Avgerou and Cornford, 1998). IS use technology such as computer hardware and software, to provide an effective and efficient way of processing data and transforming it into a variety of information products (Turban *et al*, 1996).

2.2.2 Social and organisation issues in information systems

Despite attempts to make software development more rigorous, a considerable proportion of computer system development effort results in products that do not provide user satisfaction (Vidgen, 1997). Even with the availability of a wide array of advanced software development methodologies, techniques and tools, serious problems with software is still being faced. Examples include, the CONFIRM reservation system (Oz, 1994), Australian government IS project - Mandata (Sauer, 1993) and the London ambulance service computer-aided despatch system project (Beynon-Davies, 1999).

Laudon and Laudon (1995) points out that failure can be viewed from both the technical and organisational point views. From the technical point view, the major source of IS failure is inattention to quality and overall system quality, which shows itself in hardware and software faults (Avgerou and Cornford, 1998). From an organisational view point, the major causes of IS failures are: insufficient or improper user participation in systems development process, lack of management support, poor management of the implementation process and high levels of complexity and risk in systems development (Laudon and Laudon, 1995).

Many authors have identified a number of social and organisational issues that affect software systems success (Le Quesne, 1988; Poltrock and Grudin, 1994). These can be summarised in the three level behavioural model proposed by Curtis *et al* (1988) as consisting of individual level, group level and organisational level. Palvia (1998) and Enns and Huff (1999) have suggested the inclusion of the environmental level, which are broad categories of factors such as economic, government, technological and cultural. Figure 2-1 presents some of these issues as identified in the literature (note that some factors overlap at different levels).

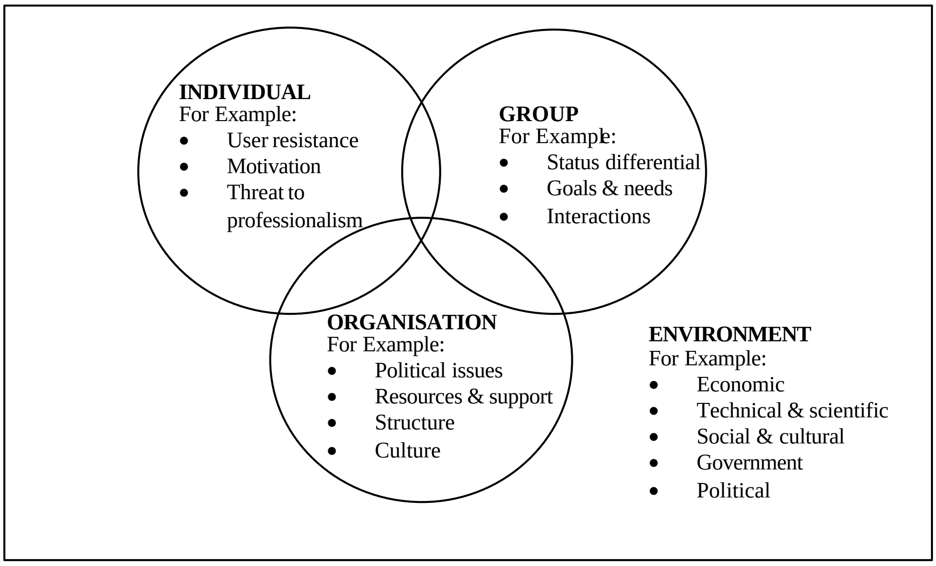


Figure 2-1. Organisational factors affecting software systems.

a. Individual behavioural factors

Organisations are made up of individual members who may be stakeholders in the system being developed. Information systems should address both the demands of the organisation and the needs of the individual that use the information technology to perform their jobs (Avgerou and Cornford, 1998). Where the needs of the individual and the demands of the organisation are incompatible, this can result in frustration and conflict. The important factors identified in literature at individual level include:

- *User resistance.* User resistance to change is perceived by many IS professionals as the primary reasons why information systems fail (Lyytinen and Hirschheim, 1987). User resistance is attributed to innate conservatism, lack of felt need, uncertainty, lack of involvement in the change, lack of management support and poor technical quality (Hirschheim and Newman, 1988; Somerville and Rodden, 1995).

- *Motivation issues.* Many authors consider motivation as an important factors that contributes to the success of IS (Le Quesne, 1988; Grudin, 1994). For example, some systems such as groupware can lead to activity that violates social taboos, threatens existing political structures and de-motivate users (Grudin, 1994).
- *Threat to the notion of professionalism*. These factors are related to role identity and status issues. For example, Le Quesne (1988) found that status issues were clearly apparent in individual reactions to the use of the system, where some managers felt using the system made them feel like typists.

b. Group behavioural factors

Individuals within organisations belong to one or more groups. Because of the social and political factors at work in a group settings, achieving groupware acceptance is more difficult than a single user product acceptance (Grudin, 1994). A number of factors influence IS success at the group level, including:

- *Group status differential.* Groups can be composed of individuals with different status. The presence of status differential can be detrimental to the quality of group decisions, for example they can discourage lower-status individuals from passing critical information to influential higher-status individuals (Grudin, 1994; Tan *et al*, 1999).
- *Group differing goals.* Individuals working together in a group may have different needs and objectives (goals), which are most often opposing (Somerville and Rodden, 1995). These goals depend on their responsibilities and status in the organisation, their personal involvement with the organisation (e.g. they may own shares) and external circumstances. Therefore, these goals need to be harmonised.
- *Group interactions.* Group interaction and social contact are important factors contributing to the overall job satisfaction (Bjorn-Andersen, 1988). Organisational structures can inhibit timely communication between groups, for example, systems designer group and developer group (Curtis *et al*, 1988). Therefore, when designing information systems, it is necessary to make sure that the possibilities for social interaction are enhanced.

c. Organisational behavioural factors

Individuals and groups interact within the structure of the formal organisation. However, people sometimes relate in an informal manner and there are factors at the organisation level that can affect the success of the software system. These include:

- *Political issues.* Many authors state that organisational power and politics are important factors influencing the successful development and implementation of an information system (Sauer, 1993; Avgerou and Cornford, 1998).
- *Organisational resources and support*. Organisations work overload, skill shortage and budgetary pressure can affect the success of software system (Lynex and Layzell, 1997). Therefore, it is important to secure management support to allow them to allocate adequate resources for software development projects (Sauer, 1993).
- *Organisational structure.* Organisational structures and processes can also hinder the successful application of good and acceptable design principles resulting in poor design features (Poltrock and Grudin, 1994). Lynex and Layzell (1997) found that in some organisations the structure encouraged people not to co-operate or share strategic information and instead promoted competition amongst the business units and this inhibited software reuse.
- *Organisational culture.* Many authors point out that cultural factors have significant effect on IS and software success (Woherem, 1992b; Somerville and Rodden, 1995). In the context of software processes, cultural factors influence the introduction of new processes and the modification and evolution of existing approaches to software development (Somerville and Rodden, 1995).

d. External environmental factors

The organisation functions as part of the broader external environment of which it is part. The environment affects the organisation through, for example, technological and scientific development, economic activity, social and cultural influences and governmental actions (Sauer, 1993; Palvia, 1998; Enns and Huff, 1999). This in turn will affect the software development and IS success. The identified factors include:

- *Economic environment.* The economic environment of a country has an impact on the success of IT implementation because the economy determines such things as the availability of funds to purchase hardware, software and technical support

(Enns and Huff, 1999). For example, DCs do not have the financial resources to improve their infrastructure and invest in IT (Okot-uma, 1992).

- *Technological and scientific development environment.* The existence and sophistication of technology infrastructure is an important factor impacting on the success of IS, especially in DCs (Janczewski, 1992).
- *Social and cultural environment.* Cultural factors are significant at two levels, first at the organisational level where organisations develop their own distinctive culture which is recognised and (generally) accepted by the people working in these organisations and then at the national level where different countries have different cultures (Somerville and Rodden, 1995).
- *Government and regulatory environment.* The government and regulatory environment has impact on the IT and IS success (Enns and Huff, 1999). For example, Tallon and Kraemer (1999) provide the Ireland's IT success as a result of government initiative to encourage overseas investments and multinational corporations.

The next section briefly discusses the approaches adopted for developing information systems and addressing the problems identified in this section.

2.2.3 Approaches to information systems development

There are a number of approaches to building information systems such as the social-technical approaches (Mumford, 1990), software engineering (Sommerville, 1995), web approach (Kling, 1996). For example, the web approach emphasizes on computing infrastructure such as electricity, communication lines, physical space, and people skilled in using and maintaining them. These approaches can be roughly classified into the hard and soft methods (Avison and Fitzgerald, 1995; Flynn, 1998). Hard methods are biased towards the technical issues, such as software engineering (Sommerville, 1995), and soft methods are biased social issues such as social-technical approaches (Mumford, 1990).

The objective of software engineering is to produce software systems that are delivered to a customer with the documentation which describes how to install and use the system (Sommerville, 1995). The driving concern of the engineering approach is the development of a complex technical system, achieving efficiency, and producing error-free systems (Avgerou and Cornford, 1998). This goal is realized through project activities that are well planned and disciplined, taking a top-down, reductionist, approach to decompose the inherent complexity of information handling into smaller achievable verifiable tasks (Avgerou and Cornford, 1998). COTS-based systems development approach, presented in chapter 3, is an example of this approach. The software engineering approaches have been criticized because they often do not pay attention to the human and social issues, and also neglect the wider organisational context (Avison and Fitzgerald, 1995; Flynn, 1998).

The social-technical approach is aimed at developing a system that consists of both the human subsystem and technical subsystem in an integrated manner. Unlike the software engineering approach, the basic principle of the social-technical approach is that the IS development process is an intervention in an organisation, intended to improve the way people communicate with each other and do their jobs (Avgerou and Cornford, 1998). The social-technical approach is said to be an important strategy for addressing organisational and social issues (Mumford, 1990). The social-technical approach is more fully discussed in the next section.

2.2.4 Social-technical approaches to information systems

This section will introduce the social-technical systems (STS) theory and discuss the application of this theory to information systems development.

a. Social-technical systems theory

The goal of the STS is to develop a system that promotes a work environment in which people can perform effectively their organisational role and can achieve personal development and satisfaction (Avgerou and Cornford, 1998). It is one that recognises the interaction of technology and people and produces work systems which are both technically efficient and have social characteristics, which lead to high job satisfaction (Mumford, 1990). This development strategy has its origins in studies from the 1950s on the relationship between the social structure and technology in organisations (Trist, 1978). The argument for a social-technical approach is that if designers ignore the human subsystem (that is attitudes, skills, preferences, habits and physical capacities of the workers, their interaction patterns, and their social skills) then the very best technical system will be under-utilised (Banner and Gagne, 1995).

The four major social-technical subsystems are the personnel subsystem, technological subsystem, environment subsystem and the self-managed work teams. The personnel or social subsystem refers to human resources and human capital assets, which work in the organisation and the totality of their individual and social attitudes (Sena and Shani, 1999). The technological subsystem of an organisation consists of the tools, techniques, procedures, skills, knowledge, and devices used by members of the personnel system to accomplish organisational tasks (Trist and Bamforth, 1951). The business environment (environmental subsystem) is composed of elements in the marketplace in which the organisation competes (including customers, competitors and host of other outside forces) (Sena and Shani, 1999).

STS theory has evolved into a set of fairly stable and recognisable propositions (Pasmore and Sherwood, 1978). These specify (1) that the design of the organisation must fit its goals; (2) that employees must be actively involved in designing the structure of the organisation; (3) that variances in production or service must be controlled as close to their source as possible; (4) that subsystems must be designed around relatively whole and recognisable tasks; (5) that support systems must be congruent with the design of the organisation; (6) that a high quality of work life should be provided; and (7) that changes should continue to be made as necessary to meet environmental demands.

Banner and Gagne's (1995) review of STS reveals a bias toward variety, challenging jobs, social support, collaboration and recognition, whole jobs (rather than division of labour), minimised external controls, and performance feedback on a timely basis. This supports the principles advocated by Trist (1981). For example, that the social-technical work group is self-regulating (internally) rather than externally regulated by supervisors and that variety and complexity are valued over the simplicity and routineness in work. Table 2-1 encapsulates the differences between the traditional and social-technical view of organisations.

Traditional View	STS View
People are extensions of machines	People are complements to machines
People as expendable spare parts	People as a resource to be developed
Maximum task breakdown	Optimum task grouping
Narrow skills	Multiple broad skills
Autocratic management style	Participative management styles
Competition	Collaboration

Table 2-1. Difference between traditional and STS view (Banner and Gagne, 1995)

b. Application of STS theory to information systems

In information systems, the STS approach conveys a mixture of practical, ethical and theoretical concerns (Avgerou and Cornford, 1998). The practical concern is that information technology-based systems, even if they are well designed, often fail to bring the desired benefits that they are intended to produce. The ethical concerns embody a fundamental value of industrial democracies, that workers should have a share of the benefits brought by new technology, such as the betterment of human conditions. From the theoretical point of view, the social-technical approach conveys the awareness that information systems development is a political process, which may redistribute organisational power.

Many researchers have tried to apply the STS in information systems development and implementation (Bostrom and Heinen, 1977; Pava, 1983; Mumford, 1990). For example, Mumford (1990) applied some of the principles of STS theory to develop the ETHICS method, which is oriented towards the introduction of organisational systems incorporating new technology. ETHICS incorporates the joint philosophies of participation (see section 2.2.4c) and the social-technical design (see section 2.2.4d). Flynn (1998) points out that ETHICS addresses the human and wider organisational issues in the design and development of the system through user participation and the social-technical approach assists in ensuring that the system does not address an incorrect problem.

Another approach that has been influenced by the STS theory is Multiview (Avison and Wood-Harper, 1990). Multiview is based on social-technical solution to developing information systems and uses user participation approach of ETHICS methodology. Multiview is aimed at addressing questions related to the organisation as a whole, the people working in the organisation, the particular aspect of the human-computer interaction, the various functions that the information system is to carry out and the technical specification for performing those functions (Avison and Wood-Harper, 1990). Multiview is very flexible and has been used in many situations, for example, it has been adapted for IS planning and development for DCs (Bell, 1996) and Internet-based IS development (Vidgen, 1999).

Participation and STS design are perceived to be important concepts when applying STS theory to information systems (Mumford, 1990; Avgerou and Cornford, 1998).

c. Participation

Customer participation refers to the behaviours and activities of the customers during information system development while customer involvement refers to the participation in the system development process by representatives of the target user group (Ives and Olson, 1984; Emam *et al*, 1996). It is argued that participation may lead to increased user acceptance by developing realistic expectations about the systems capabilities, providing an arena for bargaining and conflict resolution about the selected product and leading to system ownership by users (Keen, 1981; Ives and Olson, 1984; Mumford, 1990; Taylor and Felten, 1993). It is further argued that participation improves employee satisfaction and productivity, and that involvement provides the content of, and the reasons for, empowerment (Mumford, 1990; Taylor and Felten 1993; Axtell *et al*, 1997).

Customer participation may vary from direct, where all parties are affected by the system are involved, to indirect, where employee representatives serve on decision-making committees (Ives and Olson, 1984). For example, in Joint Application Development (JAD) a variety of proprietary and custom developed methods are used for conducting workshops in which users and technical developers work together on IS project planning, requirements definition, user interface design or other activities (Davidson, 1999). Participative Design (PD) encourages even stronger user involvement than JAD, emphasises mutual learning (between users and designers) and much less structured meetings (Wood and Silver, 1995). Carmel *et al* (1993) claims that JAD can lead to increased IS quality, reduced development costs and reduced systems development life-cycle time.

However, it has been argued that participation may not improve information system quality and success (Ives and Olson 1984; Flynn, 1998). For example, Axtell *et al* (1997) identified a number of problems when integrating user participation into software development, these include the problem of the relationship between users and developers, lack of senior management support and the difficulty in accessing users that are massive and widely distributed. Avgerou and Cornford (1998) criticise participative approaches because they are time-consuming and expensive processes, which tend to lead to compromises regarding systems design. Nevertheless, participation is regarded as an effective strategy of improving software design outcomes and as a means of incorporating human and organisational aspects such as the design of jobs, work processes and usability (Gould *et al*, 1991; Bravo, 1993; Axtell *et al*, 1997). Therefore, participation has been incorporated in the social-technical framework for COTS software selection discussed in section 6.5.

d. Social-technical analysis and design

Another consequence of the STS theory in information systems is the perception of the inevitable interaction between the social and technical subsystems and the need to incorporate both in the design of systems (Bostrom and Heinen 1977; Mumford, 1990). Social-technical analysis involves specifying the social and technical objectives separately, and later merging them into the social-technical (see figure 2-2). The social objectives is aimed at improving job satisfaction and quality of working life while the technical objectives are aimed at improving business efficiency.

Job satisfaction is defined as the fit between an individual or group's job needs and expectations and the requirements of the job which they presently occupy (Mumford, 1990). The ETHICS job satisfaction framework covers three broad areas (a) needs associated with personality, including knowledge needs and psychological needs, (b) needs associated with competence and efficiency in the work role, including efficiency factors such as support services and systems of work control and tasks needs, and (c) needs associated with employee values, in particular ethical needs such as how employees wish to be treated by management.

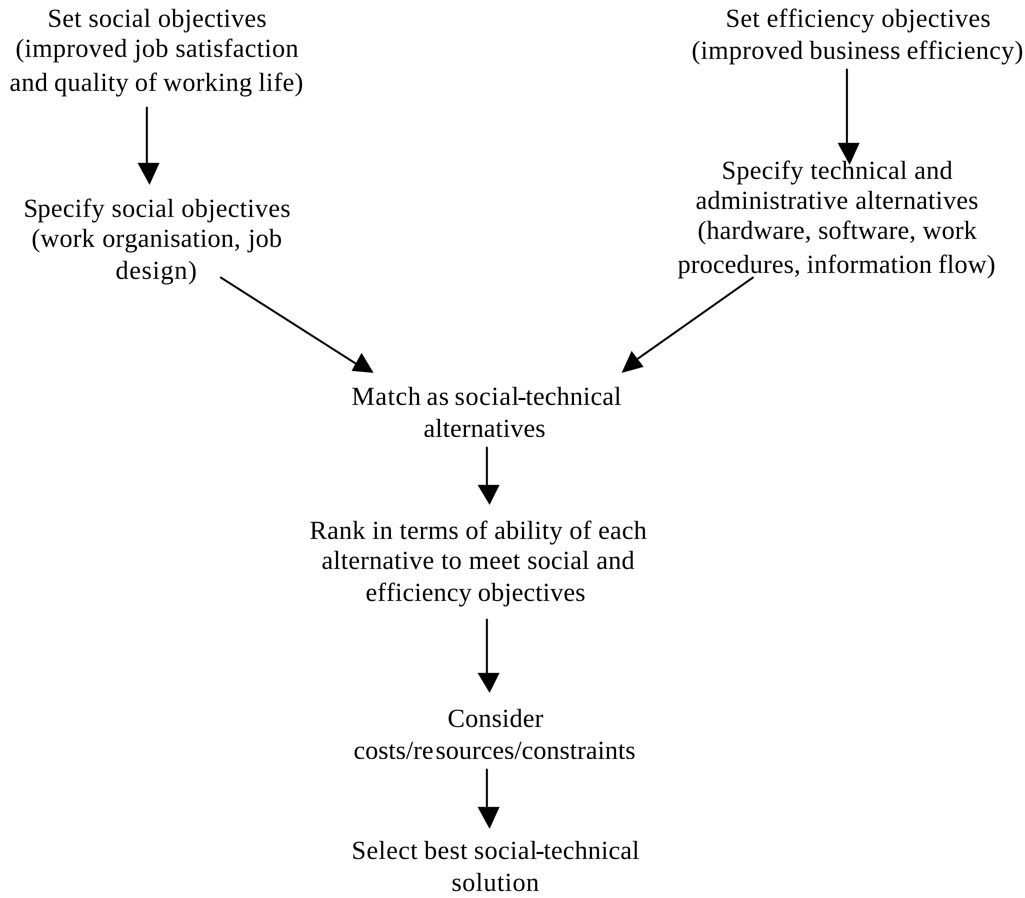


Figure 2-2. Social-technical Systems Design (Mumford, 1990)

Social-technical analysis is useful for incorporating the social (non-technical) objectives of the system, thus ensuring that the proposed system addresses the correct problem (Mumford, 1990; Flynn, 1998). Therefore, this technique is recommended in the social-technical framework for COTS software selection (see section 6.5) to decompose the high level requirements into social criteria and technical criteria.

The next section will discuss the development and implementation of IS in developing countries.

2.3 Information systems in developing countries

This section begins by discussing the common characteristics of DC. Then, reviews IS development in DCs and problems restricting their successful implementation. The section also examines current approaches adopted by DCs to implement IS, including their dependence on international aid organisations. Lastly, the section presents information systems in a specific developing country, Zambia.

2.3.1 Common characteristics of developing countries

Developing countries differ markedly in terms of size of country, endowments of resources, nature of industrial structure and levels of per capita national income (Ingham, 1995). This poses the difficulty of coming up with an agreed definition of a developing country. For example, Bell (1996) points out that although the term "developing country" is often used to describe nations seeking power over their own affairs, the term has no conclusive definition. However, the World Bank (2000) uses the term "developing country" is applied to low and middle income economies as well as economies in transition from central planning. This definition is based on classification of the economies of the world according to income; low-income (GNP per capita of less than $760), middle-income (GNP per capita between $761 and $9,360) and high income (GNP per capita more than $9,361) (World Bank, 2000).

However, most researchers argue that though DCs are diverse and at varying stages of economic development they have several common characteristics (Nafziger, 1990; Ingham, 1995; Todaro, 1997; Roy, 1999). It is argued that in this "diversity there is basic unity" that gives them a shared identity and reason to work together for common object of reducing poverty and underdevelopment (Roy, 1999). The common characteristics include:

- Low levels of living, comprising low incomes, high inequality, poor health and inadequate education (Ingham, 1995; Todaro, 1997; World Bank, 2000).
- Low levels of productivity (Ingham, 1995; Roy, 1999). Todaro (1997) argues that to raise productivity in DCs, domestic savings and foreign finance must be mobilised to generate new investment in physical capital goods and build up the stock of human capital (e.g. managerial skills) through investment in education and training.
- High rates of population growth (Ingham, 1995; Todaro, 1997), for example in 1990 the average population growth in DCs was 2.1 percent per year compared to 0.5 in developed countries (Nafziger, 1990). This rapid population growth has

contributed to a rapid growth in the labour force and increasing urban unemployment in DCs (Nafziger, 1990).

- High and rising levels of unemployment and underemployment. Todaro (1997) suggests that one of the factors contributing to the low levels of living in DCs is their relatively inadequate or inefficient utilisation of labour in comparison to the developed countries.
- Significant dependence on agricultural production and primary product exports. Most low-income countries are predominantly peasant agricultural societies, however they do not run agriculture business enterprises being more concerned with survival (Nafziger, 1990). DCs also depend significantly on primary products (food, raw materials, minerals, organic oil and fats) for their exports.
- Dependency and vulnerability in international relations (Ingram, 1995; Todaro, 1997). For many DCs, a significant factor contributing to the persistence of low levels of living and rising unemployment is the highly unequal distribution of economic and political power between rich and poor nations (Todaro, 1997). For example, the colonial transfer of inappropriate structures (educational, health and public administrative systems) and the international brain drain (the migration of experienced and trained personnel to developed countries for better conditions of service).

It is these shared characteristics that distinguish them from other countries. Therefore, following Todaro (1997) this thesis adopts a definition of “developing countries” as those countries of Asia, Africa, the Middle East and Latin America, which are characterised by low levels of living, high rates of population growth, low income per capita and general economic and technological dependence on the developed countries. Heeks (1999) uses the term “developing country” to encompass not merely the nations of Africa, Asia, Latin America and the Caribbean, but also the transitional economies of Eastern Europe and other nations on the European economic periphery such as Turkey and Ireland.

In section 2.3.5, the characteristics of Zambia are discussed indicating how it meets enough of the criteria presented in this section to qualify as an example of a developing country. The next section discusses information systems development in DCs.

2.3.2 Developing information systems in developing countries

There are a number of models and approaches proposed by researchers for developing and implementing information systems for DCs. For example, in implementing IS in Chile, Robey *et al* (1990) describe the following steps: systems design and development, hardware installation, selection and training of system administrators, clerical training and testing, evaluation and redesign. Bell and Wood-Harper (1990) propose a methodology based on a modification of Multiview approach (see section 2.2.4b). Bell (1996) argues that Multiview is an appropriate analysis and design tool for use in DCs because the eclectic nature of the approach proves effective and this eclecticism makes it possible to relate the methodology to other effective development approaches such as the rapid rural appraisal. Rapid rural appraisal focus on the needs of the context in which planning and analysis take place and is said to be appropriate for systems planning in the resource-poor context of many DCs.

However, Bell (1996) found that Multiview was not sensitive enough for local needs of a developing country context and that it stops at the software design stage leaving the other activities of the software life cycle. Therefore, Multiview was improved on during its application in the developing country context in terms of its sensitivity, expressed in the explicit appraisal of the analyst's assumptions and development of tools that were responsive to local needs. Furthermore, an additional stage was introduced to deal with software and hardware selection, training and implementation issues. This example illustrates the point made by Janczewski (1992) and Avgerou (1996) that ISD practices and methods are not universal and need to be adjusted to the socio-economic, cultural and organisation setting.

There are a number of factors that are important for successful implementation of IS in organisations in DCs such as availability of adequately trained and experienced personnel to use the information systems (Walsham *et al*, 1990). Mohan *et al* (1990) stresses the importance of creating awareness in top management about the value of the proposed systems and the importance of educated and well-trained personnel to use the system being implemented. Therefore, it important to develop training guidelines and implementation procedures based the user requirements and knowledge gained from the analysis and design stage (Bell, 1996). Other important techniques recommended for DCs include the use of a critical success factors to identify important problems (Mohan *et al*, 1990; Averweg and Erwin, 1999), participation (Bell and Wood-Harper, 1990; Bell, 1996), socio-economic (Mursu *et al*, 2000) and social-political perspectives (Walsham *et al*, 1990).

Mohan *et al* (1990) points out that since one aspect of the evidence of the value of a system is usage, it is important that initial applications selected for implementation must have short development time periods. Therefore, the prototyping method where the system evolves through an iterative process based on the testing the prototype with users and modifying it on the basis of their feedback, would be appropriate for DCs. Calhoun and DeLargy (1992) argue that applications in DCs should be computerised incrementally to allow the implementation experience to influence the design and to increase organisational fit and commitment. Avgerou and Land (1992) argue that implementation of a new system comprises both technical and socio-organisational changes. This suggests the appropriateness of applying the social-technical approaches to systems development and implementation of IT in DCs.

The next section attempts to discuss and classify some of the problems that impinge on successful implementation of IT in DCs.

2.3.3 Problems of developing information systems in developing countries

Although a variety of problems are faced by DCs, this review focuses on those problems that are unique to the DCs and which may have a significant impact on their assimilation of IT. This review highlights the following problems: skilled human resources deficiency, economic constraints, systems infrastructure deficiency, and social cultural issues and applications problems.

a. Skilled human resources deficiency

The lack of skilled human resources is agreed as being the principal barrier blocking the diffusion and efficient/effective exploitation of IT systems and is at the root of the problems which DCs face (Bogod, 1979; Woherem, 1992a; Corr, 1995). There is clearly a problem of quantity and quality (mismatch between needs of industry and trained personnel) of IT personnel (Bhatnagar, 1992a). Bell and Wood-Harper (1990) observed that in DCs the lack of skills does not only refer to professional IT skills but also to the management skills required to plan, co-ordinate and manage the introduction of technology. Okot-uma (1992) argues that there is a lack of computer maintenance expertise in a number of DCs, shortage of skilled personnel for the operation of available computers, a shortage of application design expertise and a scarcity of programmers.

A number of factors have been isolated as prime causes of these deficiencies, such as the evolution of technology, high turnover of skilled staff due to poor conditions of service, lack of counterpart training under technical assistance, etc. (Okot-uma, 1992). Jere (1992) attributes this to lack of adequate computer education and training to institutions at national level. While Corr (1995) suggests that these problems are as a result of the general economic conditions prevalent in these countries and the lack of appropriate government policies. Researchers have made a number of proposals to mitigate against these problems, including the following:

- Implementing technical education and training which can create and mature the IT professionals (Ojo, 1992). For example, adequate exposure of students in tertiary and higher institutions to computer education to make them potential supporters of IT applications in their future work places.
- Encouraging regional co-operation between DCs so that methodologies and approaches developed in one country and that are applicable to other countries, can be used (Bhatnagar, 1992a).
- Encouraging governments and organisations in DCs to improve the salaries of IT personnel and other conditions of service (Woherem, 1992a). For example, encouraging an enterprise culture within educational institutions can result in a degree of financial independence leading to reduced dependence on donor organisations and ability to improve lecturer’s pay (Corr, 1995).
- Encouraging organisations to develop and adopt strategies for planning IT manpower and skills acquisition (Woherem, 1992a). For example, organisations with similar manpower/skills shortages and future needs might come together and act as a pressure group on the government.

However, these proposed solutions assume that organisations have the resources and supporting infrastructure to implement them, but this is often not the case for DCs (Bogod, 1979; Okot-uma, 1992; Bhatnagar, 1992b).

b. Economic constraints

Economic constraints is another set of major obstacles restricting the application of IT in DCs, including the non-existence of reliable background statistical information, inadequate capital to finance IT, etc (Okot-uma, 1992). Several DCs suffer from both a lack of financial resources and a limited domestic market (Janczewski, 1992; Prevost and Gilruth, 1997). These countries import IT because they lack an indigenous IT industry and scarcity of foreign currency forces them to depend on donor agencies for much of their IT imports (Bhatnagar, 1992b).

The cost of setting up a technological development path for DCs and catching up with developed countries would be close to impossible and economically prohibitive (Bogod, 1979). There would need for substantial capital investments in computing, electronics and telecommunications industries, research and educational establishments. Janczewski (1992) found that the total fund requirements for the average IT investment are quite substantial in comparison with the cost of labour and this was encouraging companies to promote a manual operation rather than invest in computers. What DCs need are systems that can be supported within the existing infrastructure and economy capacities. DCs, for example, can contribute to the software technology with minimal capital investment by developing indigenous software that can be sold on international market (Bhatnagar, 1992b).

c. Systems infrastructure deficiency

Successful implementation of computer systems is dependent upon there being a systems infrastructure on which to build. Okot-uma (1992) points out that in DCs the electrical power utility has been intermittent and the inconveniences caused have not been negligible. Janczewski (1992) also indicates that the quality of the power supply in most Western African nations is poor because of overloading and the low quality of the power grid. Low quality electrical power supply may be a result of climatic or economic conditions existing in these countries as well as the legislation protecting power authorities (Janczewski, 1992).

In addition, DCs do not have adequate telecommunication infrastructure; telecommunication networks are not totally reliable and transmission rates are still slow (Okot-uma, 1992; Prevost and Gilruth, 1997). This is a problem in DCs because investment in telecommunication are expensive and, together with low GNPs, such investments are more difficult (Janczewski, 1992). This requires government intervention and there is little that can be done by the individual organisation implementing IT. Some researchers, such as Woherem (1992b), suggest that institutions in DCs should band together and pressure governments to improve the infrastructure.

d. Socio-cultural issues and application problems

Woherem (1992b) argue that the methods and techniques from developed country are most often inappropriate to DCs because they generally do not take into consideration social-cultural contexts of DCs. The socio-cultural factors that may impede the implementation of IT in DCs include language, cultural attitudes toward speed and time, protection of cultural identity and the meaning of authority (Robey *et al*, 1990). Similarly, Ojo (1992) also identified a number of socio-cultural and organisational issues when applying IT in Nigeria. These include: culture of self-motivated commitment to public services, culture of over-politicised decision making, culture secrecy, culture of bureaucratic complexity and culture of exploitative IT vendoring. Therefore, it important to adapt techniques and systems from developed countries when transferring them to DCs (Bjorn-Anderson, 1990; Janczewski, 1992).

Bhatnagar (1992b) argues that the factor contributing to the low impact and penetration of computers in these economies is the type of use that such computers have been put to. Most DCs have used their computers for routine transaction processing tasks rather than strategic information systems. Bogod (1979) points out that the priority areas of application of computer systems in DCs are different from developed countries. This has implications on the transferability of software. For example, DCs have the following problems and tasks: development and exploitation of natural resources, raising educational standards for the population, raising the standards of health, and increasing food production (Bogod, 1979; Heeks, 1995). See section 2.3.4b for more detailed discussion of technology transfer.

2.3.4 Strategies promoting information systems in developing countries

The strategies discussed in this section include international aid, technology transfer and government policies.

a. International aid organisation and donors

Foreign aid encompasses all official grants and concessional loans, in currency or in kind, that are broadly aimed at transferring resources from developed to less developed nations on development or income distribution grounds (Todaro, 1997). Many international organisations - such as the USAID, the various UN agencies, the World Bank, ODA, to name but a few - have represented an important source of funding for projects involving IT in DCs (Odedra, 1995).

Foreign assistance agencies are involved in the introduction of computer technology in several ways, for example as project components, commodity import programs and informatics agencies such as World Resource Institute (Daly, 1992; Odedra, 1995; Prevost and Gilruth, 1997). However, very rarely is any investment or donation based on analysis of the requirements of the recipient organisation or country (Janczewski, 1992; Odedra, 1995). In cases where the IT systems is donated in the form of gifts, it is less meaningful to talk about feasibility studies since the usual procedure is to get the equipment first, find the recipient later and then suggest a possible implementation approach (Janczewski, 1992). Furthermore, computer systems are often given to organisation with no training or provision for extra recurrent costs (Odedra, 1995).

Most of the foreign aid that DCs receive is tied aid, because it limits the receiving country to purchase goods and services from the donor-countries (Todaro, 1997). Tying foreign aid effectively limits the range of technological options in aided projects (Janczewski, 1992; Odedra, 1995; Todaro, 1997). DCs usually accept equipment which may not be the most suitable, or of the highest quality, and which may rule out any attempt to standardise uses of equipment as a way of reducing the range of spare parts needed (Odedra, 1995).

The donor initiatives have also tended to have a narrow sector-specific focus often to the exclusion of broader national contexts (Gyamfi-Aidoo *et al*, 1995). They have ignored the context and do not provide guidelines (best practices) to the DCs. Overall, support from international organisations has tended to be sporadic and uncoordinated, and often creates great problems for countries in terms of compatibility and parts (Odedra, 1995). Often, the consultants hired may not be familiar with the country they are working in, they may not know about the social or cultural aspects of the nation, nor of the organisational issues (Odedra, 1995). Greater awareness about these problems should help funding agencies and recipient countries develop coherent strategies (Bjorn-Andersen, 1990). For example, donors need to conduct a comprehensive analysis of the receiving organisation's problems and requirements before providing the technology.

b. Technology transfer

Cross-border technology transfer is a process where knowledge on production processes is acquired by entities within a country from sources outside the country (Fransman, 1986). Bihari and Varner (1994) suggest that technology transfer only work if there is a real existing market for the technology and requires educating the recipients. Alford (1994) point out that successful technology transfer requires the researcher to show the immediate benefit of the technology being transferred. Fowler (1994) further emphasises partnering as an essential ingredient of effectiveness of cross-border technology transfer and maintains that the facilitator must be knowledgeable about both the technology and its target organisation. Lindgaard (1994) argues that technical, human resource issues, and strategic plans should be considered in planning technological changes or diffusion in work places.

In transferring technology from developed to DCs, it is necessary to consider aspects of prior research on introduction and assimilation of new technologies and on implementation of IT (Fowler, 1994). In addition, it requires that attention be paid to understanding the misalignments derived from technology, which has been developed according to a different set of contextual elements (see figure 2-3). Examples of these elements include political factors, socio-cultural factors, resources and operational environment (Specter and Sahay, 1992; Vozikis *et al*, 1992; Janczewski, 1992; Fowler, 1994). People may resist new technology because it brings about many social changes, creating new social gaps and class structures. Fear of unknown and the possibility of loosing a job are potentially important reasons for resisting new technology. Therefore, it is important to understand organisational cultures as well as technology before beginning any technology transfer, and deciding which technologies are good candidates for transfer (Pfleeger, 1999).

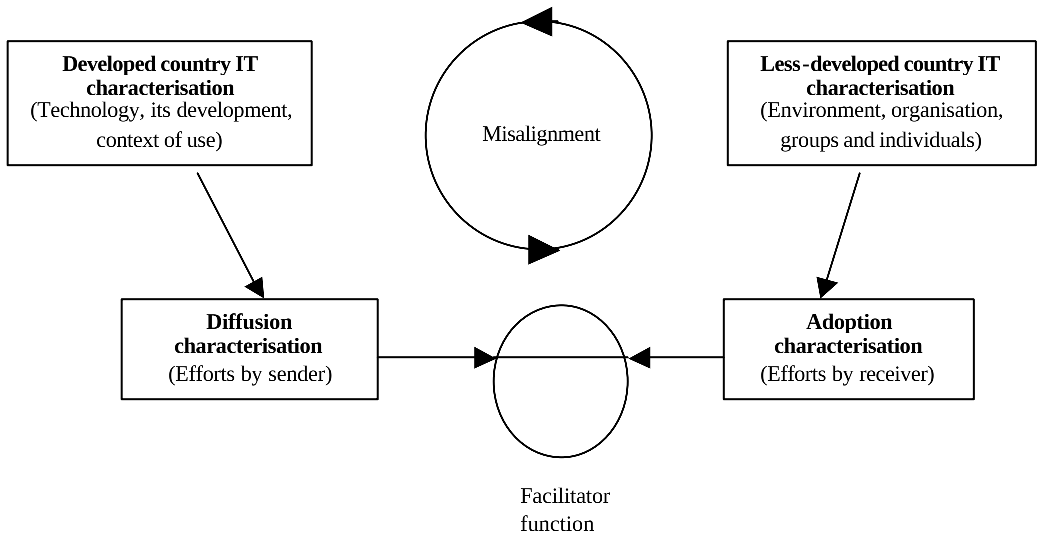


Figure 2-3. Critical dimension in cross-border transfer of IT (Fowler, 1994)

A fundamental difference between DCs and developed countries is that governments in DCs often exert considerable influence over the industries and organisations by controlling, for example, access to key resources and setting costs and prices (Montealegre and Applegate, 1994). Scarcity of financial resources, inadequate infrastructure, cultural and resource constraints limit technology transfer and the assimilation of IT in DCs (see section 2.3.3). Therefore, it important to be aware of these issues when transferring technology to DCs.

c. Government policies

Hassan (2000) argues that in DCs government policies and economic conditions play a critical role in shaping the development of a software organisation. Tax laws, foreign currency regulations and legal system (especially intellectual property protection laws) are examples of some of these factors. Heeks (1999) provides the example of Israel and Taiwan that have used a raft of tax breaks, marketing subsidies, grants, loans, legislative updates and reduction of red tape in an effort to achieve this.

Therefore, a supportive national environment can positively influence the introduction and assimilation of IT, for example a strong government promotions, co-ordinated approach to infrastructure development and political stability (Montealegre and Applegate, 1994).

Robey *et al* (1990) argue that the transfer of most commercial technology in DCs is accomplished through multinational corporations (MNCs). Therefore, MNCs assist in economic development directly through the import of technology and economic development in turn has cultural and socioeconomic ramifications. Tallon and Kraemer (1999) provide an example of Ireland economic success through overseas investment and MNCs through IT production. IT production created employment opportunities and key technical and managerial expertise which provided an alternative to emigration. Irish workers in MNCs also developed skills that enabled them to form their own businesses. However, reliance on overseas investment and MNC renders DCs vulnerable to shifts in global demand for IT. For example, in the late 1997 the Seagate's Irish plant was closed because of over-capacity in the disk drive industry and fallout in the Asian markets (Tallon and Kraemer, 1999)

Ojo (1992) and Corr (1995) points out that formulation of appropriate and well-articulated national IT policy can contribute to solving some of the problems that plague DCs. The IT policy can be used by the government to define the urgent IT problems and sets goals and direction for human resource development, procurement and maintenance of equipment, improvement of communication infrastructure and establishment of local computing industry (Jere, 1992; Odedra, 1993b; Corr, 1995). Singapore provides an excellent example of how strong government sponsorship, a co-ordinated approach to infrastructure development and the development of national IT plan can positively influence the introduction and assimilation of IT (Montealegre and Applegate, 1994).

Education and training can help to promote IT development in DCs (Heeks, 1999). For example, Ireland’s success in IT is because they have heavily invested in people, with half the population going into tertiary education and considerable emphasis on IT courses. Training in DCs should not just be limited to technical skills but also managerial and policy skills (Bell and Wood-Harper, 1990). Therefore, there is need for training of managers and policy makers in DCs so that they have an appreciation of the technical aspects of IT, allied to an understanding of their use in organisations and their socio-political implications (Walsham *et al*, 1990)

2.3.5 Information systems in Zambia

This section provides some background information about Zambia, a developing country where some of the studies were conducted.

a. Characteristics of Zambia

Zambia is a land locked nation with a population of approximately 9 million in 1997 and a land area of 752,610 square miles. There are three major cities of Lusaka (capital), Ndola and Kitwe. Zambia is divided in nine provinces and adopted a multi-party democracy in 1991. Zambia is a former British colony that achieved its independence on 24th October 1964. English is the official language.

A number of characteristics of Zambia indicate that it is an example of a developing country (see section 2.3.1 for characteristics of DCs). For example, in 1998, 73% of the country's total population was considered poor, out of which 58% were extremely poor and 15% moderately poor (CSO, 1998). The incidence of poverty was higher in rural areas compared to urban areas. Furthermore, in 1998 the estimated annual population growth rate was 3.1%, one of the highest in the world, implying approximately a 23-year doubling time of the population (UNDP, 1998). With regard to education levels, 27% of the country's population never went to school in 1998 while only 0.2% had a bachelor degree and above. The UNDP (1998) report indicates that three quarters of those in grade 6 in 1998 were functionally illiterate due to a marked decline in the country's quality of primary education.

Zambia's economy is characterised as diversified by largely agriculture and mining, nearly 40 percent of GDP comes from mining or mining related manufacturing (see table 2-2). Zambia exports depend mainly on primary products including copper, cobalt, zinc, lead, cement, tobacco and sugar. Agriculture was far the largest employer in 1998, employing 67% of the country's total employed persons, out of which 59% were male and 76% female. The unemployment rate in Zambia was estimated at 50%, indicating the inadequate or inefficient utilisation of labour. Zambia's economy suffers from a very high debt service burden, heavy reliance on a single export product (copper) and a history of excessive public sector direction of, and direct participation in, the production of goods and services (World Bank, 1998).

(% of GDP)	**1977**	**1987**	**1997**	**1998**
Agriculture	18.1	12.0	18.6	19.4
Industry	41.6	45.5	34.6	29.6
Manufacturing	19.6	28.5	13.5	12.8
Services	40.2	42.4	46.7	50.9
Private consumption	51.5	63.1	79.4	83.8
General government consumption	26.4	20.4	11.3	10.8
Imports of goods and services	41.9	35.7	37.0	38.4
Key economic ratios				
GDP (US$ billions)	2.5	2.3	3.9	3.4
Total debt/ GDP	93.1	292.5	171.9	204.8

Table 2-2. Structure of the Zambian economy (World Bank, 1998)

In 1990, Zambia undertook a structural adjustment program to counter these problems. The structural adjustment program combined trade policy reforms, deregulation, and exchange rate adjustment with stabilisation policies designed to restore fiscal and balance of payments equilibrium and price stability (World Bank, 1998). The government made a number of policy achievements as a result of this structural adjustment program. These include elimination subsidies on maize and fertiliser; decontrol of prices and exchange rates; revision of investment laws and regulations; freed interest rates; reduction of budget deficit (excluding grants and interest); and adoption of a privatisation program. However, the GDP continued to decline due primarily to reduced output in agriculture and mining. In 1999, the estimated per capita GDP of Zambia was US$309 (USAID, 2000).

b. Information and communication technology in Zambia

Most of the application for which computers have been used in Zambia can be classified as data and transaction processing, and operational and management control systems in the tertiary sector of development (Odedra, 1993a). Surveys conducted on computer utilisation and staffing in Zambian industry revealed that there is a small expanding use of computers based largely on mainframes and minicomputers but a growing number of PCs (Corr, 1995). In Zambia the gap between the demand for computer expertise and the available supply is large and growing. It was found that computers were mainly used for accounting and business related functions with COBOL strongly represented. Furthermore, that there is a considerable difference between the application of computers in the industrially developed countries of the north and DCs such as Zambia. It is in the ability to address the specific needs of the local industry that the advantage of the indigenous training course lie.

There are three public Internet Access Providers (IAP) in Zambia (AIS, 1999). The larger is one of the first in Africa is the University of Zambia's ZamNet, with about 3000 subscribers. The second largest IAP is CopperNet which was created out of ZCCM's privatised IT division and has about 2700 subscribers. The third is the Zambia Telecommunications Company (ZAMTEL), which is the sole provider of basic telecommunication services in the country.

The major problem in Zambia is that there is a critical shortage of IT manpower, especially of information analysts, who can understand both the technical aspects of IS and the behavioural, social, political and organisational aspects (Jere, 1992). This is because there is no computer technology awareness until the tertiary level or at the place of work (Jere, 1992; Corr, 1995). Most of the institutions conducting courses are not well equipped to teach IT. Another major problem affecting the pace of information technology is the scarcity of foreign currency to import hardware and software (Shitima, 1990). This has led to poor infrastructure, for example most users of data communication lines often experience difficulties especially in the rainy season. There is no hardware manufactured or assembled in the country though there is a lot of in-house software development (Odedra, 1993a).

A number of strategies have been recommended to address these problems. This include establishing or revamping the existing institution charged with formulating national policies on training, planning, procurement, co-ordination etc. of information technology (Shitima, 1990). For example, revising IT curriculum at Evelyn Hone College to include social-political issues and CASE tools (Jere, 1992). Developing a national IT policy (Corr, 1995) and considering the viability of assembling locally some of the hardware (PCs) as well as encouragement of software development for exports (Shitima, 1990). Although these recommendations are viable in the long term there is need for short-term interventions. Therefore, it is important to elicit a better understanding of the current situation, practices and problems regarding software development in Zambia. This would result in development of better ways to help them develop and implement information systems more effectively.

2.4 Summary

This chapter reviewed relevant literature on the social-technical approaches to information systems and information systems in DCs. The chapter began by introducing the field of information systems and discussing the organisational and social issues affecting successful development and implementation of information systems. A framework based on Curtis *et al* (1988) and Palvia (1998) was used to classify these factors in human level, group level, organisational level and environmental level.

The common approaches to information systems development were introduced. The chapter was then concerned with the social-technical approaches aimed at addressing these human, social and organisational issues. A brief description of the social-technical systems theory and its background was given. The application of social-technical theory to information systems was discussed, highlighting the concepts of participation and STS design. Review of literature formed the basis for developing the field studies and the development of a social-technical framework for COTS software selection discussed in section 6.5.

The chapter then examined common characteristics of DCs and adopted the classification of developing countries based on Todaro (1997). The problems of developing information systems in DCs were presented including skilled human resources deficiency, economic constraints, systems infrastructure deficiency, and social cultural/ applications problems. A number of strategies for addressing these problems and developing information systems in DCs were reviewed. The strategies reviewed are dependence on international aid organisations, technology transfer and government policies. The characteristics of a specific developing country, Zambia, were also discussed.

This review indicates that there are still problems of developing information systems in DCs because of lack of resources. Therefore, what DCs require are methods or techniques that can help them reduce costs associated with information systems development and maintenance while at the same time take advantage of the new technology. A COTS-based systems development offers such benefits, including reuse across projects. The next chapter will discuss COTS-based systems, their benefits and risks.

3 COTS-Based Systems

This chapter reviews the field of COTS-Based Systems (CBS) focusing on evaluation and selection of COTS software components. The CBS development process, as well as the benefits and risks associated with CBS are discussed. The COTS software evaluation problems, process, method and techniques are discussed in this chapter. The chapter also presents a review of important frameworks for evaluating and selecting COTS components.

3.1 Introduction

Software engineering is a discipline that integrates process, methods and tools for the development of computer software (Pressman, 2000). The IEEE (1993) defines software engineering as the application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software; that is, application of engineering to software. Therefore, the challenge for software engineering is to produce high quality software products that meet the user requirements with a limited amount of resources and within a certain time schedule.

Software products can be said to fall into two broad classes namely bespoke (customised) and generic (packaged) products. Bespoke products, are systems that are commissioned by a particular customer and a specific contractor develops the software specifically for that customer (Sommerville, 1995). These are expensive because all the development cost has to be met by a single client. Generic products (also known commercial-off-the shelf (COTS) and commercial software) means all software sold as tradable product (purchased from a vendor, distributor or store) for all computer platforms including mainframes, workstations and microcomputers (Sawyer, 2000). These are cheaper because their development costs are spread across hundreds of different customers (Sommerville, 1995).

According to Oberndorf (1997) the term “COTS” is meant to refer to things that one can buy, ready-made, from some manufacturer’s virtual store shelf (e.g., through a catalogue or from a price list). It carries with it a sense of getting, at a reasonable cost, something that already does the job. The scenario of developing unique system components is replaced by the promise of fast, efficient acquisition of cheap (or at least cheaper) component implementations. Examples of COTS products include Geographic Information Systems (GIS), Graphical User Interface (GUI) builders, office automation, email and messaging systems, databases and operating systems (Vigder *et al*, 1996).

This chapter will introduce COTS-based systems development to provide the context in which the research is conducted and then focus on COTS software evaluation and selection. The chapter also reviews a number of important frameworks developed for COTS software evaluation and selection (see section 3.4). These frameworks are specific to COTS software evaluation and selection. Therefore they are different from generic social-technical frameworks for information systems such as Multiview and ETHICS discussed in section 2.2.4. The objective of this review is to identify what is missing in COTS software evaluation and selection frameworks as well as understand characteristics that are important for a generic COTS software evaluation and selection framework.

3.2 COTS-based systems development

This section provides an overview of what COTS-based systems development is and how it differs from component-based software engineering. The section also reviews the benefits and risks associated with building systems from COTS software and discuss the potential benefits of CBS in DCs.

3.2.1 COTS and Component-based software engineering

Many authors use COTS-Based Systems (CBS) and Component-Based Software Engineering (CBSE) interchangeably (Haines *et al*, 1997; Tran *et al*, 1997; Fox *et al*, 1997). For example, Tran *et al* (1997) defines CBS and CBSE as a systematic approach to the selection, evaluation and integration of reusable software. At the foundation of this approach is the assumption that certain parts of large software systems reappear with sufficient regularity (Haines *et al*, 1997). Therefore, common parts should be written once, rather than many times, and common systems should be assembled through reuse rather than rewritten over and over.

However, it important to differentiate between CBS and CBSE. CBSE focus on integrating multiple software that are ready “off-the-the shelf” whether from a commercial source (such as COTS) or re-used from another system (Brown and Wallnau, 1996b). On the other hand, CBS focus on building systems from COTS software components either by integrating a number of them into a system or by acquiring a single working system and adapting and extending it for local needs (Vidger *et al*, 1996). Table 3-1 encapsulates the differences.

There are two distinct ways to use COTS software. In one, a single complete working COTS software system that satisfies most of the user requirements is purchased and used as platform upon which to build a system (Coppit and Sullivan, 2000). For example a database management system can be purchased and used to build a payroll system. The second CBS model is one which involves purchasing a number of COTS software components each satisfying some part of the requirements of the system and integrating these components into the required system (Tran *et al*, 1997).

CBSE	CBS
Normally focus on integrating multiple software components	Build system by integrating multiple COTS software components but can also acquire a single COTS software such as database and adapt/ extend it
Build system without necessarily using COTS software components e.g. re-use component from another system or produce component from in-house	The system must have at least one COTS software component
The developer may have the burden of upgrading and maintaining some software components e.g. in-house components	Developer is not responsible for upgrade and maintenance of COTS software components

Table 3-1. Differences between CBS and CBSE

The focus of this research is the second CBS model in which system developer buys a number of components (usually without the source code) from third party developers and then integrates them into the system. This approach is important because many systems need functions in multiple orthogonal sub-domains, each of which tends to be addressed by a different COTS software package (Coppit and Sullivan, 2000). Therefore, in this thesis CBS is defined as building software systems by integrating pre-existing software components that are ready “off-the shelf” from a commercial source (i.e. COTS) (Brown and Wallnau, 1996b; Vidger *et al*, 1996). In CBS the developer is not responsible for ongoing support or release of updates to the COTS software and maintenance (Vigder *et al*, 1996; Dean and Vigder, 1997).

3.2.2 Benefits and risks of COTS-based systems

This section discusses the benefits and risks associated with building systems from COTS software.

a. Benefits of COTS-based systems

Building systems from COTS software offers the opportunity to lower costs by sharing them with other users, thus amortising them over a larger population, while taking advantage of the investments that industry contributes towards the development of new technologies (Oberndorf, 1997). Oberndorf (1997) argues that most organisations typically spend far too much effort on defining to the lowest level of detail the desired characteristics of systems and how the contractors are to build those systems to achieve those characteristics. Thus a lot of resources are expended developing systems and components that often already exist elsewhere. Coppit and Sullivan (2000) points out that the cost to learn and use systems built from COTS software components is reduced because people already know the particular components and the style of the components.

Clements (1996) argues that CBS has the potential to reduce the software development time because it takes less time to buy a COTS software component than it does to design it, code it, test it, debug it and document it. CBS approach can also be potentially used to reduce the spiralling maintenance burden associated with the support and upgrade of large systems since the COTS software provider is responsible for ongoing support and maintenance of the COTS products (Haines *et al*, 1997).

Braun (1999) claims that the use of standard products can help provide a common user “look and feel”; support system interoperability; shorter and more predictable development schedules; and reduced training and infrastructure costs. Building systems from COTS software also promotes competitive marketplace, thus enabling system integrators to have a wide range of choices (Szyperski, 1998). According to Voas (1999), the industry believes that using COTS software to build systems provides instant productivity gains, reduces time to market, reduces costs, fulfils organisations mandate such as US Federal Department of Defence and provide a philosophy similar to that for building hardware systems. Boehm and Abts (1999) supports these views and further suggests that COTS software provides rich functionality (see table 3-2).

Advantages	**Disadvantages**
Immediately available: earlier payback	Licensing, intellectual property procurement delays
Avoids expensive development	Up-front license fees
Avoids expensive maintenance	Recurring maintenance fees
Predictable, confirmable license fees and performance	Reliability often unknown or inadequate; scale difficult to change
Rich functionality	Too-rich functionality compromises usability, performance
Broadly used, mature technologies	Constraints on functionality, efficiency
Frequent upgrades often anticipate organisation's needs	No control over upgrades and maintenance
Dedicated support organisation	Dependence on vendor
Hardware/software independence	Integration not always trivial; incompatibilities among vendors
Tracks technology trends	Synchronising multiple-vendor upgrades

Table 3-2. COTS advantages and disadvantages (Boehm and Abts, 1999)

b. Risks and costs associated with COTS-Based Systems

Building systems from COTS software components poses a number of risks, for example lack of support if COTS provider goes out of business (McDermid, 1998; Braun, 1999). Use of COTS products makes an organisation dependent upon the product vendor (Boehm and Abts, 1999). Therefore, there is the potential risk that the vendor might stop making that component or produce components that are incompatible with older versions (Clements, 1996). In addition, integrating COTS components involves negotiating and managing costs, as well as tracking licenses to ensure uninterrupted operation of the system (Haines *et al*, 1997). For example, a license expiring in the middle of a mission might have disastrous consequences.

Another risk associated with CBS is instability due to periodic releases of COTS software. Rapid product release cycles imply that the use of components will be schedule-driven, possibly at the expense of product and system stability (Carney and Wallnau, 1998). Remaining with the older versions of COTS products might cause interoperability problems with upgrades to other systems, for example, the upgrade to the operating system (Fox *et al*, 1997). Furthermore, it is typically harder and more expensive to certify COTS products than their bespoke counterparts - as they were not typically designed for certification (McDermid, 1997). COTS products may be cheaper to buy initially, but integrating upgrades involves reassessments and retrospective assessments which can be very expensive especially in safety critical systems (McDermid, 1998).

There are other risks associated with building systems from COTS software, for example COTS components may not meet the performance parameters (Braun, 1999). Their use may make the system operate too slowly, may have capacity limits that are constrained by customer needs and may not provide flexibility to easily adjust these. Furthermore, maintaining systems that incorporate COTS component can be difficult and labour intensive for several reasons, for example due to incompatible upgrades (such as added features or bug fixes) (Voas, 1999). In addition, using systems that incorporate COTS components poses security risks; for example a component may contain intentional security flaws such as viruses (Lindqvist and Jonsson, 1998).

However, despite the above risks organisations are migrating to CBS development because of the increasing availability and reliability of both generic and domain specific reusable COTS software components (Tran *et al*, 1997). In addition the emergence of component technology that support interoperability of COTS products is one of the key driving factors for interest in CBS (Brown and Wallnau, 1996b; Allen and Frost, 1998). Examples of component technology include Common Object Request Broker Architecture (CORBA) and Component Object Model (COM) and Enterprise JavaBean component system (Pressman, 2000)(see section 3.2.4d).

3.2.3 Potential benefits of COTS-based systems in developing countries

As has been argued in section 2.3.3, DCs lack the infrastructure and resources to successfully build and implement information systems (Okot-uma, 1992; Janczewski, 1992; Prevost and Gilruth, 1997). Furthermore, there is pressure for organisations in DCs to become self-reliant and reduce dependence on foreign aid (Worehem, 1992b; Todaro, 1997; Odedra, 1995; Corr, 1995). Building systems from COTS software components offers the opportunity to lower costs by sharing them with other users and has potential for reduced training and infrastructure costs (Oberndorf, 1997; Braun, 1999; Boehm and Abts, 1999). Therefore, by employing CBS, DCs will not spend too much time on developing overly expensive and inflexible systems, with only one customer to bear the development and maintenance costs over the life of the component, being unable to easily capitalise on advances in new technology.

Developing CBS is becoming feasible due to the increase in the quality and variety of COTS products such as operating systems, databases, email and messaging systems, GIS, office automation software and GUI builders (Vigder *et al*, 1996; Haines *et al*, 1997). This list comprises applications and components that are mature and pervasive in a large number of systems both in developed countries and DCs (Shitima, 1990; Woherem, 1992b; Prevost and Gilruth, 1997). The number of available components continues to grow and their quality and applicability continue to improve. Therefore, organisations in DCs are turning to CBS because of these benefits.

A number of socio-cultural factors impede on technology transfer and implementation IT in DCs (Robey *et al*, 1990; Ojo, 1992; Montealegre, 1994). It has been emphasised in literature that software systems developed with different socio-cultural context should be adapted when applied in DCs (Bjorn-Andersen, 1990; Janczewski, 1992). CBS offers the capabilities of extending and tailoring COTS software products through APIs, plug-ins and scripting languages (Vigder *et al*, 1996; Vigder and Dean, 1997). Collins and De-Diana (1992) provide an example of how commercial educational software developed in other countries was adapted and proved useful in Mexico notwithstanding the various technical, social/cultural and organisational constraints. Therefore, CBS can be applied to DCs and also deal with some of socio-cultural factors.

3.2.4 COTS-Based Systems development process

CBS can be partitioned into the following essential activities: requirements engineering, component qualification, component adaptation, component integration and component update (for systems evolution) (Brown and Wallnau, 1996b; Tran *et al*, 1997; Maiden and Ncube, 1998)(see figure 3-1).

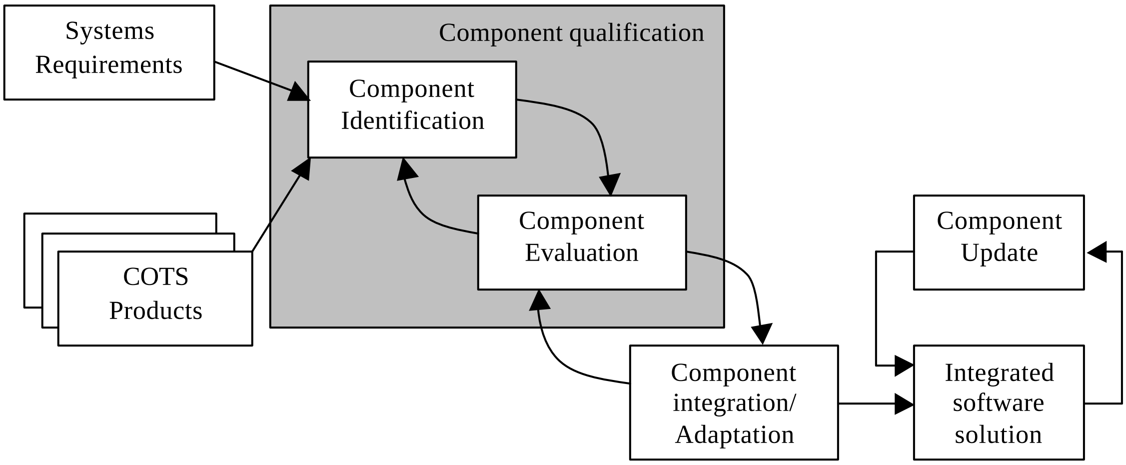


Figure 3-1. CBS systems development model (adapted from Tran *et al*, 1997)

a. Requirements engineering

Requirements engineering covers all of the activities involved in discovering, documenting and maintaining a set of requirements for a computer-based system (Sommerville and Sawyer, 1997). It is an iterative process comprising of requirements elicitation, requirements analysis, requirements specification and requirements validation (Pohl, 1997). In CBS, the goal of this activity is to determine requirements which will assist in establishing a basis for evaluating and selecting appropriate COTS software candidates (Dean and Vigder, 1997). A number of techniques are used to elicit requirements including traditional techniques (e.g. interviews, surveys), group elicitation techniques (e.g. brainstorming, JAD), prototyping, cognitive techniques and contextual techniques (Nuseibeh and Easterbrook, 2000).

The use of COTS products introduces new problems for requirements engineers, for example, deciding when to acquire new customer requirements and when to reduce the number of candidate products (Maiden and Ncube, 1998). Dean and Vidger (1997) propose acquiring a small number of high-level requirements prior to an iterative and concurrent product evaluation and selection. Finkelstein *et al* (1996) suggest that the acquisition process should focus on requirements that can be used to distinguish most between COTS products. Therefore, in CBS the requirements engineer should concentrate on eliciting high-level requirements because stakeholders often have prior knowledge of candidate products that they prefer. Furthermore, this saves resources by not spending too much time in describing in detail the requirements of the system.

Tran *et al* (1997) argue that the requirements should be broken down and organised into collections of domain-specific requirements. This is important to support the early identification of candidate COTS products for evaluation as well as early identification of subsystems that cannot be supported by COTS products. In order to realise the benefits of COTS software, Vigder *et al* (1996) suggest that a procurement process must be in place that defines requirements according to what is available in the marketplace. This is contrarily to the traditionally procurement process, which identifies strict requirements which either excludes the use of COTS components, or requires large modifications to COTS packages in order to satisfy the requirements. However, it is important that requirements are not defined so specifically that only one particular COTS product is suitable (SEL, 1996).

There are several methods for modelling requirements in literature. The "classical" methods include data modelling, functional modelling, behavioural modelling and object-oriented modelling (Pohl, 1997; Pressman, 2000). Many authors recommend scenarios and use cases techniques for specifying the requirements of COTS software systems because they simplify the mapping between requirements and the proposed COTS (or reusable) packages (Jacobson, 1995; Maiden and Ncube, 1998). However, these methods emphasise the technical issues while neglecting the equally important social issues (Jirokta and Goguen, 1994). Therefore, there is potential for applying social-technical approaches discussed section 2.2.4 to support the RE process for CBS and address these non-technical issues.

b. Component qualification

Component qualification (also known as COTS software evaluation) is more fully discussed in section 3.3. Component qualification ensures that a candidate component will perform the functionality required, will properly "fit" into the architectural style specified for the system, and will exhibit the quality characteristics (e.g. performance, reliability, usability) that are required (Pressman, 2000). CBS success depends on successful evaluation and selection of COTS software components and was therefore the main focus of this research.

c. Component adaptation

Because individual components are written to meet different requirements and are based on differing assumptions about their context, components must be adapted when used in a new system (Haines *et al*, 1997). Furthermore, COTS component that have been qualified for use within an application architecture may exhibit conflicts, for example inconsistent resource management (such as memory, swap space and printers) (Pressman, 2000). Therefore, component adaptation is used to mitigate these conflicts. This usually involves some form of wrapping – locally developed code that provides an encapsulation of the component to mask unwanted and incompatible behaviour (Brown and Wallnau, 1996a).

There are a number of techniques used for adapting and extending systems. These include modifying source code (where the supplier can be requested to modify the source code); plug-ins; application programming interface or API (these are component internal programming language used to add functionality to the component); and scripting (Vigder et al, 1996; Haines et al, 1997). A plug-in registers with the COTS system of its capabilities and services and the system calls the plug-in as required, for example browsers use plug-ins to enhance their functionality to display more types of image formats. A script can be used to extend the behaviour of a component (by having the component execute the script), or it can be used as a co-ordination mechanism to integrate two or more components (by providing the "glue" for linking the components together).

It has been emphasised in literature that software systems developed with different socio-cultural context should be adapted when applied in DCs (see section 2.3). Therefore, the capabilities for extending and tailoring COTS software products through APIs, plug-ins and scripting languages suggests that CBS can easily be used to adapt systems for DCs context.

d. Component integration/assembly

Component integration phase consists of developing the software mechanisms for interconnecting the selected COTS products; developing necessary enhancements to these products and integrating and testing the final system (Tran *et al*, 1997). There are several architectural styles and mechanisms for integrating COTS components including procedure calls (e.g. a procedural library); filters and data exchange model; data sharing (e.g. data repository); messaging system - message passing; and underlying object model (e.g. Microsoft COM model). Filters and data exchange model are mechanisms that allow users and applications to interact and transfer data such as cut-and-paste in office automation (Pressman, 2000). In the message passing approach, components communicate by passing messages informing other components of their actions and the requesting services from other components through a message broker such as CORBA (Brown and Wallnau, 1996b).

There are standards that have been developed with the goal of achieving component integration, the most popular are CORBA (OMG, 1998), COM (Microsoft, 1999) and Enterprise JavaBeans (EJB)(Sun, 1998). The COM model encompasses two elements: the COM interfaces, implemented as COM objects and a set of mechanisms for registering and passing messages between COM interfaces. CORBA provides an infrastructure allowing objects to communicate independent of the specific platforms and techniques used to implement the addressed objects (Fan *et al*, 2000). The EJB enable the integration of a bean (component) into a container environment defined outside the Java, thus providing a framework for integration (Szyperski, 1998).

CORBA creates no reference implementations and depends on vendors for actual delivery; this leaves a huge lag between what CORBA has standardised and what vendors are actually delivering (Sessions, 1998). On the hand, Microsoft creates COM implementations and also controls the underlying operating systems, which permits great efficiencies. However, COM technology is not portable to other platforms. The problem with EJB is that it is new and still immature (Sessions, 1998). Therefore, when building systems by integrating software it is important to evaluate and select the component integration technology suitable to the organisation.

e. Updated components (for system evolution)

As with any system CBS must evolve over time to fix errors and to add new functionality (Brown and Wallnau, 1996b). System evolution is not a simple plug-and-play approach (Haines *et al*, 1997). Systems are constantly changing, and COTS components within systems are constantly changing. This evolution of systems and their components has an impact, in a number of ways, on the maintenance of systems (Vigder *et al*, 1996). The replacement of one component with another is often a time consuming and arduous task since the new component must be thoroughly tested in isolation and in combination with the rest of the system (Brown and Wallnau, 1996b). Wrappers must typically be re-written and side effects from changes must be found and assessed.

One of the greatest challenges of maintaining CBS is the problem of what to do if a COTS vendor goes of business or fails to support the product (Voas, 1999). There is also the potential risk that the vendor might stop making that component or produce components that are incompatible with older versions (Clements, 1996). A potential solution to this problem is escrow a copy of the source code so that the user assume maintenance responsibility (Braun, 1999).

The next section discusses COTS software evaluation and selection, which was the main focus of this research. It reviews COTS software evaluation and selection problems, evaluation process, methods and techniques as well its relationship with multicriteria decision making techniques.

3.3 COTS software evaluation and selection

COTS software selection, also known as component qualification, is a process of determining "fitness for use" of previously-developed components that are being applied in a new system context (Haines *et al*, 1997). Component qualification is also a process for selecting components when a marketplace of competing products exists. Qualification of a component can also extend to include qualification of the development process used to create and maintain it (for example, ensuring algorithms have been validated, and that rigorous code inspection has taken place) (Brown and

Wallnau, 1996b). This is most obvious in safety-critical applications, but can also reduce some of the attraction of using pre-existing components.

Carney and Wallnau (1998) define a framework for COTS software evaluation that consists of four basic principles. They define COTS evaluation as 1) a form of decision making; 2) must accommodate uncertainty; 3) has a basis in design theory, especially in the relationship between the component and the system that uses it; and 4) must be situated: particular candidate products for use in the systems must be evaluated against specific evaluation criteria. Carney and Wallnau (1998) argue that the first three principles focus on the conceptual basis of COTS evaluation and are rooted in the realities of the commercial marketplace. The fourth principle deals with the necessary specificity of any COTS evaluation effort, that it must always be a particular rather than a generalised activity.

There are currently three strategies to COTS evaluation: progressive filtering, keystone identification and puzzle assembly (Oberndorf *et al*, 1997). Progressive filtering is a strategy whereby a component is selected from a larger set of potential components. This strategy starts with a large number of candidates set of components, progressively more discriminating evaluation mechanisms are applied in order to eliminate less "fit" components (Maiden and Ncube, 1998). In keystone selection strategy, a keystone characteristic such as vendor or type of technology is selected first before selecting the COTS products (Walters, 1995). Often, interoperability with the keystone becomes an overriding concern, effectively eliminating a large number of other products from consideration. The puzzle assembly model begins with the premise that a valid COTS solution will require fitting the various components of the system together as a puzzle and applies an evolutionary prototyping technique to build versions that are progressively closer to the final system (Oberndorf *et al*, 1997).

3.3.1 Problems with COTS software selection

CBS success depends on successful evaluation and selection of COTS software components to fit customer requirements (Maiden and Ncube, 1998). Successful selection of COTS software to fit requirements is still a problem because of a number of reasons. These include the following:

- *Lack of well-defined process.* Most organisations are under pressure to perform and therefore do not use a well-defined repeatable process (Kontio, 1996). The evaluators may not have the time or experience to plan the selection process in detail and therefore, they may not use the most appropriate methods in the selection process (Kontio, 1996). The resulting urgency means that evaluation decisions become pressured and a difficult decision becomes even more risky (Powell *et al*, 1997). Furthermore, when the selection process is not defined, it is reinvented each time, it is performed inconsistently and learning from previous cases is difficult (Kontio, 1996).
- *"Black box" nature of COTS components.* Lack of access to the COTS internals makes it difficult to understand COTS components and therefore evaluation is harder (Vigder *et al*, 1996). Sometimes even the supporting documentation for these components is incomplete or wrong. The design assumptions of the component are unknown; there is no source code when it needs debugging; and testing will be necessarily incomplete, since testing is only done for those functional capabilities that the customer care about (Carney and Wallnau, 1998).
- *Rapid changes in the market place.* The component user has little or no control over COTS product evolution (Vigder and Dean, 1997). Frequent releases of COTS components and rapid changes in the market place makes evaluation difficult (Carney and Wallnau, 1998). For example, a new release of the COTS component may have a feature that is not available in the component that is currently being evaluated.
- *Misuse of data consolidation method.* A common approach to consolidating evaluation results is to use some kind of weighted sum method (WSM)(Morisio and Tsoukias, 1997). However, the WSM has been criticised because assigning weights for the criteria sometimes can be inconsistent and lead to confusion about which is the most essential customer requirements (Maiden and Ncube, 1998)(see section 3.3.4a).

However, the major problem with COTS software evaluation is that evaluators tend to focus on technical capabilities at the expense of the non-technical or "soft" factors such as the human and business issues (Clements, 1996; Powell *et al*, 1997; CMU, 1998). Boehm and Abts (1999) emphasise the importance of the accurate assessment of a COTS vendor's capability and credibility in CBS. Similarly Oberndorf *et al* (1997) highlight the usefulness defining the criteria to include such issues as vendor's time in business, responsiveness to customers and willingness to support their product. Therefore, the evaluation criteria must incorporate both technical attributes and non-technical issues such as business issues and vendor capability variables.

3.3.2 COTS evaluation process

A number of researchers and organisations have proposed process models for evaluating COTS software (for example, ISO/IEC 9126, 1991; Puma, 1999). However, most authors partition it into three phases namely: evaluation criteria definition, identification of candidate COTS products and assessment (Kontio, 1996; Carney and Wallnau, 1998). These phases are briefly discussed below.

- *Defining the evaluation criteria.* The criteria definition process essentially decomposes the high-level requirements for the COTS software into a hierarchical criteria set and each branch in this hierarchy ends in an evaluation attribute (Kontio, 1996). The criteria is specific to each COTS evaluation case but should include component functionality (what services are provided), other aspects of a component's interface, business concerns such as cost and quality aspects (e.g., reliability, portability, and usability) (ISO/IEC 9126, 1991; Tran *et al*, 1997; Carney and Wallnau, 1998).
- *Identification of candidate components (alternatives).* The identification of candidate components also known as alternative identification involves the search and screening for COTS candidate components that should be included for assessment in the evaluation phase (Carney and Wallnau, 1998; Puma, 1999). Many authors highlight a number of techniques for identifying candidate COTS software including Internet search, market surveys, attending computer fairs and shows, invitation to tender (ITT) or request for proposals (RFP), vendor promotions and publications (Rowley, 1993; Kontio, 1995; Tran *et al*, 1997).
- *Assigning measure of merit to alternatives (evaluation phase).* In the evaluation phase, the properties of the candidate components are identified and assessed according to the evaluation criteria (Rowley, 1993; Kontio, 1996). Evaluation includes the acquisition of the products to be evaluated, developing evaluation

plans, installing them, learning to use them, studying their features and assessing them against the criteria (Tran *et al*, 1997).

3.3.3 Methods and techniques for evaluation

Once the criteria are defined, the screened candidate products can be examined to observe to what extent they exhibit these or other useful attributes. The following are some of the techniques used to evaluate COTS software component:

- *Paper evaluation.* This is the process of evaluating the COTS products based on supplier data in sales brochure, technical documents, telephone conversations, web site information (Maiden and Ncube, 1998). However, Beus-Dukic and Wellings (1998) suggests that vendors claims must be viewed sceptically, therefore this technique must be used in combination other evaluation techniques.
- *Market survey.* A market survey can be made using questionnaires and interviews with vendors, trade shows, user community to compile quantitative and qualitative data about the product and vendors. Finkelstein *et al* (1996) point out that in certain circumstances, especially if the package to be bought is expensive, a request for proposal (RFP) can be issued, which enable the vendors to describe their packages in a uniform manner.
- *Experimentation.* This is a rigorous test of the product to assess its compliance with the defined criteria. The experimentation process includes the acquisition and installation of the product, design of the appropriate prototype and test plan, evaluation of product and generation of report (Tran *et al*, 1997). Carney and Wallnau (1998) stresses the importance of conducting experimentation within the operating environment (context) in which the product will be used. Maiden and Ncube (1998) recommend the use of software prototypes to assist in generating test cases for product evaluation. This especially important where the evaluator do not have prior knowledge about the candidate products or prior extensive experience generating test cases.
- *Pilot study.* A pilot study is an extended version of experimentation in which "real" data from the organisation is used in the evaluation. Brown and Wallnau (1996a) argue that it is important to demonstrate the product or technology's feasibility with a pilot project. Sledge and Carney (1998) points out that because the potential for misinterpretation and misunderstanding when presenting or

discussing a commercial product is great, hands-on evaluation of COTS products is mandatory and pilot programs are a useful way to do this.

- *Vendor analysis.* Hokey (1992) points out that the vendor must be evaluated in terms of user services (installation assistance, training services and warranty) and vendor characteristics (vendor reputation and vendor stability). Checking vendor discontinuities, such as focus shifts and change of auditor, would help in this process. Haines *et al* (1997) and McDermid (1998) argue that for safety-critical systems it is important to audit the development process that was used to develop the software including the tests carried out, conformance to standards, etc.

The review of evaluation methods and techniques suggest that it is important to use appropriate methods to evaluate COTS software successfully. Furthermore, the use of particular technique or method depends on organisational experience and resources. Therefore, it would be interesting to identify and assess which methods and techniques are useful and important in COTS software evaluation.

3.3.4 COTS software evaluation and multi-attribute decision making

Carney and Wallnau (1998) argue the COTS software selection is a form of decision making. Kontio (1996), Maiden and Ncube (1998) support this view and further point out that it is a Multiple Attribute Decision-Making (MADM) process. MADM refers to making preference decisions (for example evaluation, prioritisation, selection) over the available alternatives that are characterised by multiple, usually conflicting attributes (Yoon, 1995). The goal of MADM is (a) to help the decision maker choose the best action or alternative of those studied (a choice or selection procedure), (b) to help sort out alternatives that seem “good” among a set of alternatives studied (a sorting or segmentation procedure), and/or (c) to help rank the alternatives in decreasing order of preference (an ordering or ranking procedure) (Mollaghasemi and Pet-Edwards, 1997). According to Yoon (1995) MADM share the following characteristics:

- *Alternatives:* A finite number of alternatives, from several to thousands, are screened, prioritised, selected and/ or ranked.

- *Multiple attributes:* Each problem has multiple attributes or goals or criteria. For each problem setting relevant attributes are generated, for example, to purchase a car you may have price, gas mileage, safety and warranty period.
- *Incommensurable Units:* Each attribute has different units of measurement.
- *Attribute Weights:* Almost all MADM methods require information regarding the relative importance of each attribute, which is usually supplied in an ordinal or cardinal scale.
- *Decision matrix:* A MADM problem can be concisely expressed in a matrix format, where columns indicate attributes considered in a given problem and rows list competing alternatives.

A number of MADM techniques have been applied in software selection, the most common are weighted sum or scoring method (Williams, 1992), analytical hierarchy method (Hokey, 1992; Kontio, 1996; Maiden and Ncube, 1998) and outranking method (Anderson, 1989; Morisio and Tsoukias, 1997).

a. Weighted Sum Method

The Weighed Sum Method (WSM) or scoring method is one of the simplest and probably the most popular technique for solving multiattribute decision problems (Mollaghasemi and Pet-Edwards, 1997). The WSM is based on the multiple attribute utility theory with the following axiom: any decision-maker attempts unconsciously (or implicitly) to maximise some function by aggregating all the different points of view which are taken into account (Vincke, 1992). A score in this method is obtained by adding contributions from each alternative and since two items with different measurement units cannot be added, a common numerical scaling system such as normalisation is required to permit addition among attributes values (Yoon, 1995). The total score for each alternative then can be computed by multiplying the comparable rating for each attribute by the importance weight assigned to the attribute and then summing these products over all the attributes.

The main advantage of the WSM is its ease of use and helping the decision-maker to structure and analyse the decision problem (Mollaghasemi and Pet-Edwards, 1997). However, Mollaghasemi and Pet-Edwards (1997) criticises the WSM arguing that this method tends to involve ad hoc procedures with little theoretical foundation to support it. This can lead to confusion about the most essential customer requirements (Maiden and Ncube, 1998) and make worst products on important attributes have the highest aggregated scores (Morisio and Tsoukias, 1997). Another weakness is that it is difficult to define a set of criteria and their weights as advocated in the WSM so that they are either independent of each other or if they overlap, their weights are adjusted to compensate for overlapping areas (Kontio, 1996). This suggests that WSM might not be suitable for aggregating COTS software evaluation attribute data because most COTS software attributes are not independent of each other.

b. Outranking method

Outranking methods are a class of multi-criteria decision-making techniques that provide an ordinal ranking (and sometimes partial ordering) of the alternatives (Mollaghasemi and Pet-Edwards, 1997). It has been successfully applied to COTS software evaluation and selection (Anderson, 1989; Morisio and Tsoukias, 1997). Roy (1991) developed the outranking approach and a family of evaluation methods collectively known as ELECTRE methods that are founded on the outranking relations. Yoon (1995) points out that ELECTRE methods dichotomises preferred alternatives and non-preferred ones by establishing outranking relationships. An outranking relationship (A outranks B) states that even though two alternatives A and B do not dominate each other, it is realistic to accept the risk of regarding A as almost surely better than B (Yoon, 1995).

The advantage of this approach is the ability to consider both objective and subjective criteria and the least amount of information required from the decision maker (Mollaghasemi and Pet-Edwards, 1997). Morisio and Tsoukias (1997) suggest that outranking methods are appropriate when the measurement scales of criteria are of an ordinal and when it is not possible to establish trade-offs between criteria. Mollaghasemi and Pet-Edwards (1997) points out that, although it can be expressed that alternative A is preferred to alternative B in the outranking method, it does not indicate by how much, for example with ELECTRE I a complete ranking of the alternatives may not be achieved. Therefore, this method is not appropriate for COTS software selection involving tenders that require explaining to the unsuccessful bidders why their bid was unsuccessful and how they were ranked.

c. Analytical Hierarchy Process (AHP)

AHP was developed by (Saaty, 1990) for multiple criteria decision making and has three basic functions: (1) structuring complexity, (2) measuring on a ratio scale, and (3) synthesising. AHP has been successfully applied in software and computer selection (Zviran, 1993; Kontio, 1996; Maiden and Ncube, 1998). AHP enables decision-makers to structure a multi-criteria decision making problem into a hierarchy (Yoon, 1995). A hierarchy has at least three levels; the overall goal of the problem at the top, multiple criteria that define alternatives in the middle and competing alternatives at the bottom.

AHP technique is based on pair-wise comparison between the alternatives. The result of this pair-wise comparison is converted to a normalised ranking by calculating the eigenvector from the comparison matrix's largest eigenvalue. Appendix 6 provides a worked example of the use of AHP. The advantage of the AHP technique is that it provides a systematic approach for consolidating information about alternatives using multiple-criteria (Kontio, 1996). The availability of several software packages to support the AHP has made it a popular technique (Mollaghasemi and Pet-Edwards, 1997). AHP also provides a means for measuring the consistency of the decision-maker's judgements, that is, to check the quality of the results in the comparison matrix (Zviran, 1993; Mollaghasemi and Pet-Edwards, 1997).

AHP has been criticised regarding the rank reversal: the reversal of the preference order of alternatives when new options are introduced in the problem (Dyer, 1990; Mollaghasemi and Pet-Edwards, 1997). Furthermore, that the use of a 1 to 9 measurement scale is inappropriate because of the ambiguity in the meaning of the relative importance of one factor when compared to another. However, Harker and Vargas (1990) argue that rank reversal occurs in AHP because ranking of alternatives depends on the alternatives considered, hence, adding or deleting alternatives can lead to changes in the final rank and this is consistent with rational behaviour. Furthermore, since AHP facilitates group decision-making it would be suitable for COTS software selection process that emphasises participation. In addition, AHP would be appropriate for aggregating COTS software evaluation attribute data comprising technical and non-technical issues because it incorporates both quantitative and qualitative data into the decision making process.

The next section reviews some important frameworks for COTS software evaluation and selection.

3.4 Frameworks for selecting COTS software components

A number of frameworks for evaluating and selecting COTS software components have been proposed in literature. Useful work includes Delta technology framework that help evaluate new software technology (Brown and Wallnau, 1996a) and PORE, a template based method to support requirements acquisition for COTS product selection (Maiden and Ncube, 1998). Another technique addresses the complexity of component selection and provides a decision framework that supports multi-variable component selection analysis (Kontio, 1996). Other approaches, such as the one by Boloix and Robilland (1995) focus on assessing the software product, process and their impact on the organisation. This section briefly discusses these approaches.

3.4.1 Software system evaluation framework

The Software System Evaluation Framework (SSEF) is a comprehensive evaluation framework for assessing a software system's quality and sophistication in a short time by consolidating the viewpoints of producers, operators, users, managers and stakeholders (Boloix and Robillard, 1995). The framework explicitly links process and product aspects with the ultimate utility of systems. It provides a basic set of attributes to characterise the important dimensions of software systems.

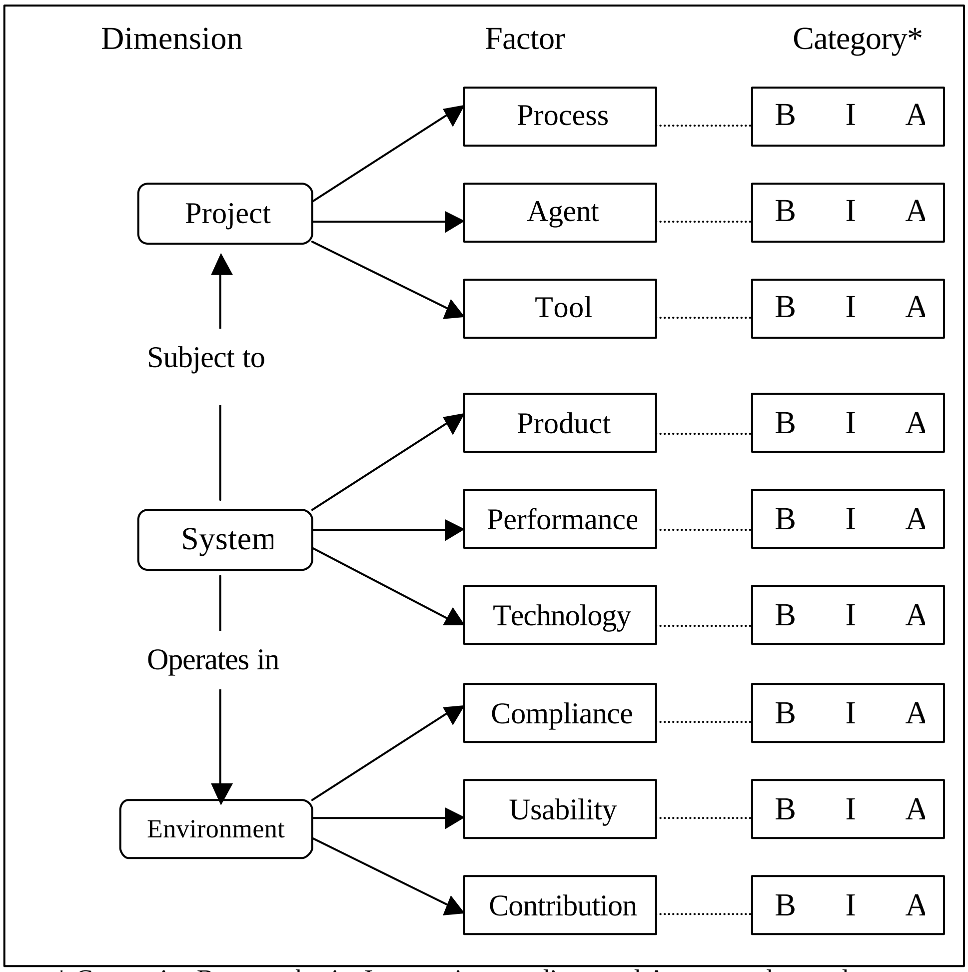
This framework proposes a top-down approach that identifies the important elements that a software system must include to foster high-level understanding. A top-down approach has the advantage of flexibility, permitting extensions by following a predefined pattern. It includes the following principles:

- *Using knowledge worlds*, the usage world records information about the (organisational) environment, the development world describes the process used to develop (or maintain) the information system, the system world describes the system at different layers of implementation detail, and the subject world comprises the subject matter (application) of the system.

- *Multiple viewpoints approach to evaluation*, user satisfaction and economic returns are important considerations for evaluating a software system's effectiveness. Organising the multitude of factors according to the different viewpoints of those producers, operators and users, facilitates the selection of important metrics for software system evaluation.
- *Defining the elements (dimensions, factors, and categories)* clearly to facilitate evaluation and reduce the evaluators' conflicting viewpoints. A top-down approach identifies the main dimensions and factors hierarchically and determines the amount of information that needs to be gathered.

The framework is organised along three dimensions corresponding to the software's producers, operators, and users (see figure 3-2). The software system information captured in the framework concerns the software product (system), its production process (project) and its impact on the organisation (environment). The project dimension characterises project efficiency considerations (i.e., process, agents and tool used to develop a system) from the viewpoint of producers and managers. The system dimension evaluates the software attributes and the type of technology implementing the software from the viewpoint of operators, system administrators and managers.

The environment dimension evaluates the level of satisfaction with the software system and the perceived contribution of the system to the organisation from the viewpoint of users and stakeholders. The environment dimension is evaluated in terms of compliance with system requirements, usability of the system from the user's perspective and contribution or benefit from the system's operation. Each criterion in the framework is categorised into three ratings: basic, intermediate and advanced. A basic category indicates the lowest maturity rating for a particular factor, an intermediate category indicates a nominal rating or industry standard, and an advanced category identifies a higher maturity rating, which signifies excellence.



* Categories B means basic, I means intermediate, and A means advanced

Figure 3-2. Software system evaluation framework (Boloix and Robillard, 1995)

According to Boloix and Robillard (1995), the framework has several applications, for example, it can provide a baseline for establishing metrics programs in organisation. The strength of the framework is that it offers a broad system snapshot by considering a number of different perspectives (end users, developers, and operators) (Brown and Wallnau, 1996b). However, this framework is not specific to COTS selection and the issues of how to define the evaluation criteria are not addressed (Kontio, 1996). The framework also gives little detailed insight into the strengths and weaknesses of a technology in comparison with its peers (Brown and Wallnau, 1996b).

3.4.2 Off-the-shelf-option framework

The Off-The-Shelf-Option (OTSO) method was developed to facilitate a systematic, repeatable and requirements-driven COTS selection process (Kontio, 1995; Kontio, 1996). It supports the search, evaluation and selection of reusable software and provides specific techniques for defining the evaluation criteria, comparing the costs and benefits of alternatives, and consolidating the evaluation results for decision making. The main principles of the OTSO method are the following (Kontio, 1996):

- explicit definitions of tasks in the selection process, including entry and exit criteria;
- incremental, hierarchical and detailed definition of evaluation criteria;
- a model for comparing the costs and value associated with each alternative, making them comparable with each other;
- use of appropriate decision making methods to analyse and summarise evaluation results.

Figure 3-3 outlines the OTSO framework. First the evaluation criteria is defined. This essentially involves decomposing the requirements for the COTS into a hierarchical criteria set. Each branch in this hierarchy ends in an evaluation attribute: a well-defined measurement or a piece of information that is determined during evaluation. A search process to identify and find potential candidates for reuse follows this. The search is driven by the guidelines in the criteria definition process. The screening is then conducted to reduce the number of candidate products and decide which alternatives should be selected for more detailed evaluation. Then after this, the evaluation process follows to evaluate the selected alternatives by the evaluation criteria and document results. The consolidation of the evaluation data is deliberately separated from producing the data to allow the use of appropriate multi-criteria decision-making techniques. The OTSO method recommends AHP for consolidating the evaluation data (see section 3.3.4 for more detailed discussion of AHP).

Brown and Wallnau (1996b) point out that evaluation is a complex task involving a combination of paper based studies, discussion with users of those components, hand-on experimentation and the OTSO framework addresses this complexity. Kontio (1996) argue that the systematic repeatable process advocated by the framework can promote learning through experience and improve the COTS selection process if it is currently conducted in an ad hoc manner. The framework's strength relies on the use of the AHP method, because it argued that AHP provides a means for measuring the consistency of the evaluation (Zviran, 1993)(see section 3.3.4). Furthermore, with AHP the level of consolidation can be chosen in order not to overflow the decision-makers with too much detailed, unstructured information. However, Maiden and Ncube (1998) found problems with implementing AHP and argue that AHP is only appropriate when there are few comparisons and when all criteria are independent.

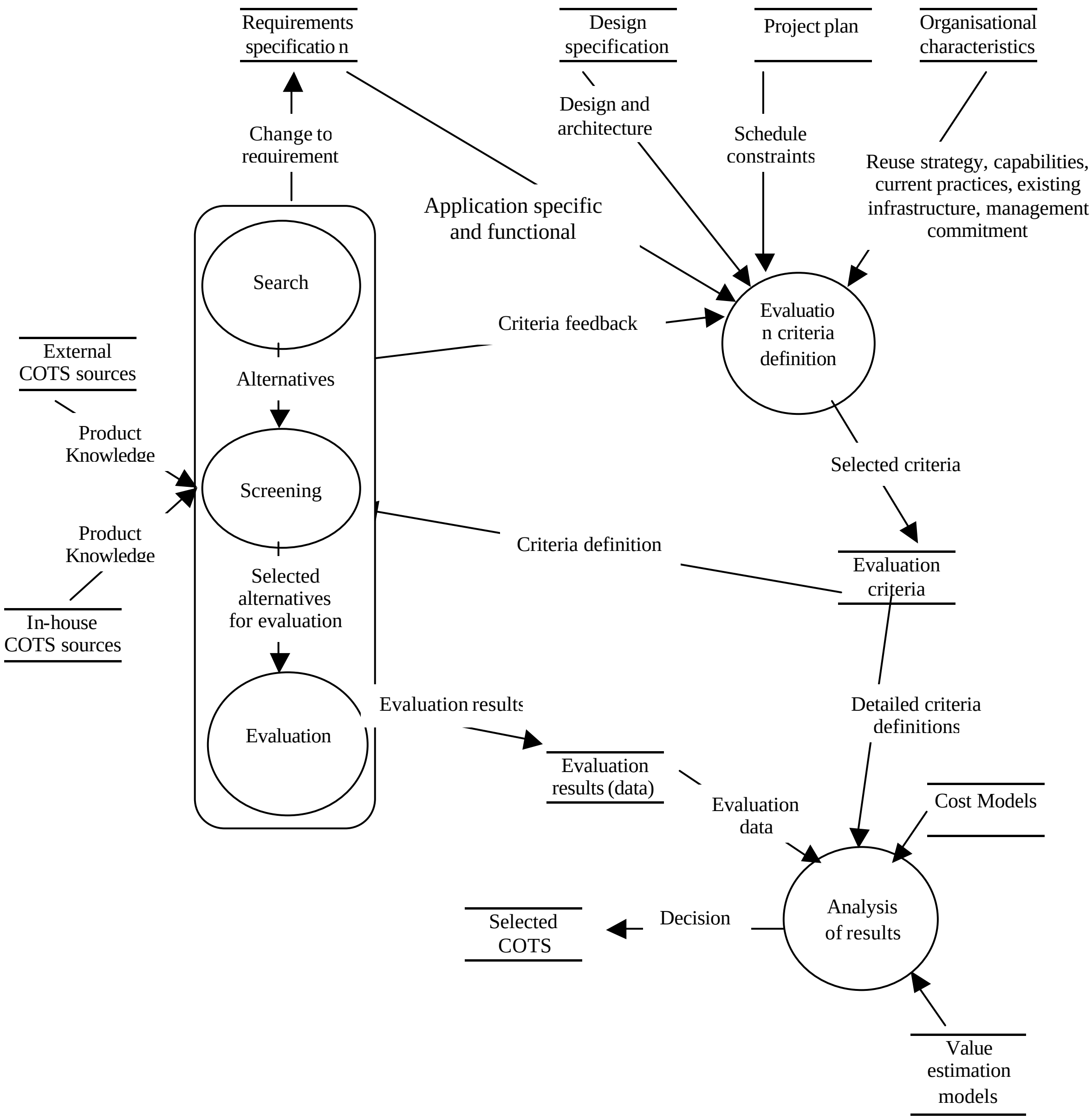


Figure 3-3. Overview of the OTSO method process

3.4.3 Delta technology framework

The Delta technology framework was developed to help companies evaluate a new software technology by examining its features in relation to its peers and competitors through a systematic approach that includes modelling and experiments (Brown and Wallnau, 1996a). The principle behind this framework is that technology evaluation depends on understanding technology "delta" descriptions of how a new technology's features differs from other technologies; and how these differences address the needs of specific usage contexts. Therefore, the Delta framework emphasises developing rigorous techniques to address both.

The framework proposes three phases to identify and then assess feature deltas namely descriptive modelling, experiment design and experiment evaluation (see figure 3-4). The descriptive modelling phase addresses feature discovery and impact prediction through the development of technology genealogies (ancestry of the technology) and problem habitats (uses of a technology, and its competitors), respectively. Experiment design phase is essentially a planning activity to formulate hypotheses about the added value of a technology that can be substantiated or refuted through experimentally acquired evidence and a set of defined experiments that can generate this evidence. Experiment evaluation phase is where evaluators conduct experiments, gather and analyse experimental evidence and confirm or refute hypotheses.

The importance of evaluating the underlying technology as opposed to the COTS product has been emphasised in literature, for example CORBA technology against COM (Oberndorf, 1997; Szyperski, 1998). Therefore, the strength of the Delta framework is the evaluation of the COTS software product underlying technology. However, Brown and Wallnau (1996a) argue that the framework can also facilitate individual product evaluations that concentrate on their distinguishing characteristics in relation to their technology precursors and product peers. While the technology Delta techniques can provide information regarding the suitability of a technology for a system, it does not address the political and economic factors that often separate a winning technology from other contenders (CMU, 1998).

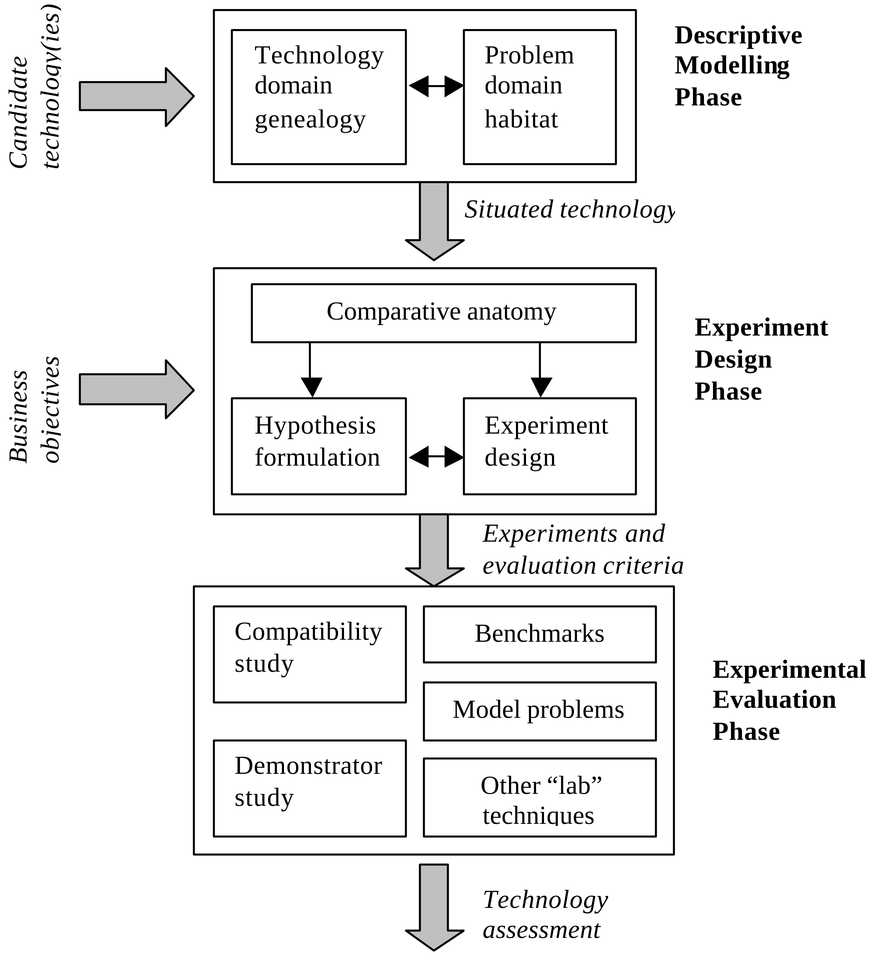


Figure 3-4. Delta technology evaluation framework

3.4.4 Procurement-oriented requirements engineering

The Procurement-Oriented Requirements Engineering (PORE) method was developed to support requirements acquisition for COTS product selection (Maiden and Ncube, 1998; Ncube and Maiden, 1999). PORE integrates requirements engineering methods and other techniques such as feature analysis (Kitchenham *et al*, 1997) and multi-criteria decision-making (Saaty, 1990). One of the main features of the PORE approach is that it encourages a requirements engineering team to acquire, describe and analyse customer requirements at the same time as acquiring, modelling and analysing candidate COTS software.

PORE provides guidelines for designing evaluation test cases and organising effective evaluation sessions through a series of templates. The first template provide guidance when acquiring essential customer requirements and product information sufficient to select and reject products as a result of supplier-given information. The second template give guidance when acquiring customer requirements and product information sufficient to select and reject products from supplier-led demonstrations using test-cases for individual requirements. The last template provides guidance to acquire customer requirements and product information sufficient to select and reject products as a result of customer-led product exploration.

PORE supports iterative requirements acquisition and product selection/rejection until one or more products are compliant with a sufficient number of customer requirements. According to Maiden and Ncube (1998), at the beginning of the PORE process there are few customer requirements but a large number of candidate products. Over time, the number of customer requirements increases and the number of candidate products decreases, as products are rejected. PORE prescribes four processes: (i) acquire information from stakeholders; (ii) analyse acquired information for completeness and correctness; (iii) use this information to make decisions about product-requirement compliance, and (iv) reject one or more candidate products as non-compliant with customer requirement (see figure 3-5).

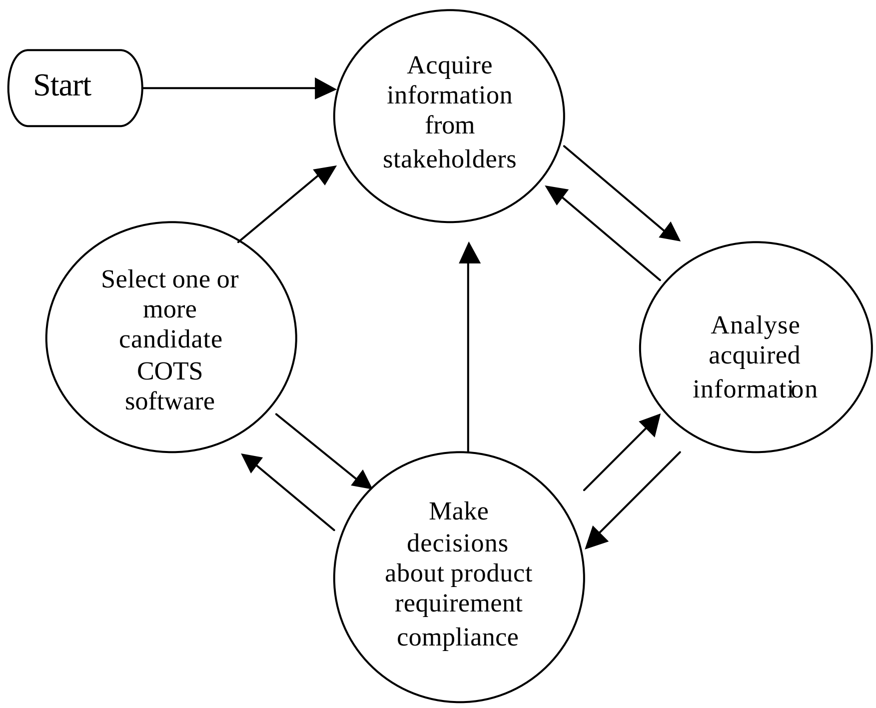


Figure 3-5. Route map showing PORE's high-level processes

The advantage of the PORE approach is that the parallel requirements acquisition and COTS software selection means the acquired requirements inform COTS software selection. Furthermore, the short-listed COTS software can inform subsequent requirements acquisition to aid further software selection. The major weakness of the PORE framework is that it is labour intensive (Ncube and Maiden, 1999), for example it is difficult to know when to stop the iterations. Furthermore, the use of the traditional requirements engineering techniques makes PORE vulnerable to neglecting the social and organisation issues as argued by Jirokta and Goguen (1994).

3.4.5 Identifying missing elements in the frameworks

The frameworks evaluated in this section provide a number of characteristics that are important for developing a generic framework for COTS software evaluation and selection. For example, all the frameworks highlight the importance of defining the requirements before selecting COTS software products. However, some frameworks (e.g. SSEF, OTSO and Delta) do not provide sufficient guidance on the process of eliciting and defining users requirements. The PORE framework which does provide such guidance, proposes the use of the traditional requirements engineering techniques making it vulnerable to neglecting the social and organisation issues.

The review also brought out the importance of defining evaluation criteria and including multiple viewpoints in software evaluation. Defining the evaluation criteria is important to ensure a well-defined repeatable process and use of appropriate methods in the evaluation (Kontio, 1996). Including multiple viewpoints would assist in incorporating the non-technical issues and reduce the risks associated with COTS software evaluation and selection. However, although SSEF advocates a multiple viewpoints approach to evaluation (user satisfaction and economic returns) it does not address how to define the evaluation criteria and incorporate multiple viewpoints in the evaluation. On the other hand, while OTSO addresses the criteria definition focus on technical issues, it neglects the equally important non-technical issues.

Evaluation of the Delta framework brought out the importance of evaluating the product underlying technology against its peers to address the needs of specific usage contexts. Most of the framework reviewed (i.e., SSEF, OTSO and PORE) do not provide techniques for evaluating the COTS software underlying technology. However, while the Delta framework provides techniques for evaluating the COTS software underlying technology, it neglects product and vendor evaluation and does not address the political and economic factors that are often important in technology selection. This highlights the neglect of non-technical issues in the Delta framework.

Another feature highlighted in this review is the importance of separating of data collection and data analysis in the evaluation to allow the use of appropriate multi-criteria decision-making techniques (Kontio, 1996). The usefulness of AHP to consolidate evaluation data was brought out in this review in that it provides structured information and a means for measuring consistency of evaluation.

Overall, what is missing in these frameworks is how to address the "soft" issues or non-technical factors such as costs, organisational issues, vendor capability and reputation (Powell *et al*, 1997; CMU, 1998). As discussed in section 2.2.2 software systems do not exist in isolation, they are used in social and organisational contexts. Neglect of social and organisational factors will lead to resistance and user dissatisfaction. Therefore, it is important for a generic framework for COTS software evaluation and selection to also address these non-technical factors. The framework proposed by this thesis aims at applying the social-technical approaches to address this problem in COTS software evaluation and selection (see section 6.5).

3.5 Summary

This chapter reviewed the field of CBS with focus on COTS software evaluation and selection. CBS was defined as building software systems by integrating pre-existing COTS software components and COTS software refers to software that one can buy, ready-made, from some manufacturer's virtual store shelf. The chapter discussed the benefits and risks associated with building systems from COTS software. For example, it is pointed out that CBS offers the benefit of reducing development and maintenance costs. Therefore, it was argued that DCs could use CBS to develop software systems cheaply by not spending too much time on developing overly expensive and inflexible systems, with only one customer to bear the development and maintenance costs over the life of the software. Furthermore, because CBS offers the capabilities of extending and tailoring COTS software products through APIs, plug-ins and scripting, it could be used to adapt software systems for DCs context.

A review of literature of research and practice on CBS suggests that evaluation and selection of COTS software components to meet requirements is problematic because of a number of reasons including rapid changes in the marketplace as well as neglect of the non-technical issues or “soft” factor. Furthermore, the existing frameworks for COTS evaluation such as the SSEF, Delta, OTSO and PORE are inadequate because they do not incorporate the non-technical factors satisfactorily. In addition, these frameworks are laborious and too complex to be adopted for general use. However, the work developed in later chapters will build on the techniques and strategies introduced in these frameworks.

The review summarised in this chapter was the basis on which most of the research was developed. For example, the development of the STACE framework and workbook used in this research drew heavily on the work reviewed in this chapter. Most importantly, insights from the discussion of COTS software evaluation and selection led to the development of the main studies and the research questions discussed in the next chapter. This research consisted of three main studies (a) eliciting current CBS practices, (b) identifying important processes and factors that support COTS software selection, and (c) evaluating the resulting framework (i.e., STACE). The next chapter presents the research method used in each of these studies.

4 Research Method

This chapter presents the research methods used in the different phases of the research. The chapter begins by describing the research design and framework and then provides an overview of the three main studies. The chapter then describes the research strategy, data collection and analysis procedures for each of the three main studies.

4.1 Introduction

The research process is the overall scheme of activities that researchers employ in order to produce research results and contribute to the body of knowledge (Nachmias and Nachmias, 1996). In this research, the process involved three main studies (a) eliciting current CBS practices, (b) identifying factors that support COTS software selection, and (c) evaluating the resulting (STACE) framework. These studies build on the literature and theoretical background as presented in chapters 2 and 3.

This chapter describes and justifies the research method followed for this research project. The research method is a way to systemise observation, describing ways of collecting evidence and indicating the type of tools and techniques to be used during data collection (Cavage, 1996). The purpose of a research method is to provide rules for communication so that the results are understood by others, thus facilitating a framework for replication and constructive criticism. The research method includes the research design, data collection and analysis procedures. The chapter first introduces the research design for the whole research and then describes in more details the data collection and analysis procedures for each of the three main studies.

4.2 Research design and framework

Research design describes the structure of research and defines concisely how all of the elements in a research project hold together. The research design is the programme that guides the investigator in the process of collecting, analysing, and interpreting observations (Nachmias and Nachmias, 1996). The elements of the research design described in this section include the purpose of research, unit of analysis and research location.

4.2.1 Purpose of research (research question)

Most DCs have yet to fully benefit from the many advances in IT because of a number of problems such as lack of resources and systems infrastructure to invest in IT (Okot-uma, 1992; Bhatnagar, 1992b; Corr, 1995). CBS offers a number of benefits that the DCs can tap into, such as reducing development and maintenance costs (Oberndorf, 1997; Braun, 1999). However, successful selection of COTS software to meet requirements is problematic because of a number of problems including lack of well-defined evaluation process and neglect of non-technical issues in the evaluation process (Kontio, 1996; Powell *et al*, 1997). Therefore, this research aim at investigating what processes (including traditional and soft factors) for evaluating and selecting software support CBS. As presented in section 1.4, the central research question of the thesis is:

> ***What processes (including traditional and soft factors) provide support for evaluating and selecting software components for COTS-based systems?***

In order to answer the above central research question and achieve the research objective a set of immediate objectives were formulated. The immediate objectives are: (a) to achieve a more comprehensive understanding of how CBS can provide support for organisations, by studying the potential benefits and risks associated with CBS, and eliciting current CBS practices; (b) to identify important processes (including traditional and soft factors) that support COTS software component selection in CBS; (c) to provide a generic social-technical framework for COTS software evaluation and selection that supports CBS.

Three major purposes of research have been identified in social sciences: exploration, description and explanation. Exploration is the attempt to develop an initial understanding of some phenomena, while description is the precise measurement and reporting of the characteristics of some population or phenomenon under study (Babbie, 1998). Explanatory research seeks an explanation of a situation or problem, usually in the form of causal relationships (Robson, 1993). The overall purpose of this research was explanatory, i.e. to explain what important processes (including traditional and soft factors) provide support for evaluating and selecting software components for CBS. The research purpose influenced the location of the research (see section 4.2.3) and selection the research strategy for each field study (see sections 4.4.1, 4.5.1 and 4.6.1).

4.2.2 Unit of analysis

Units of analysis are those things the researcher examines in order to create summary descriptions of all such units and to explain differences among them. The unit of analysis is important in the research process because making inferences about a unit of analysis on the basis of research on a unit at a different level can lead to either conceptual problems or methodological problems (Nachmias and Nachmias, 1996). Conceptual problems include lack of clarity in the research direction and focus, while methodological problems include poor external validity due to inappropriate research tools. Yin (1994) suggests that as a general guide, the definition of the unit of analysis is related to the way the initial research questions have been defined.

Babbie (1998) considers four major classes of units of analysis common in social science research. These are: (a) individuals, (b) groups, (c) organisations, and (d) social artefacts. Yin (1994) points out that previous literature in the research field provides guidance for defining the case and unit of analysis. In this study the unit of analysis is CBS social artefacts at organisational levels. The interest here is to understand the important processes (traditional and soft factors) that support COTS software selection for CBS in organisations. The selected unit of analysis influenced the research design, data collection and data analysis decisions.

4.2.3 Locating the research

The choice of research location depends on the purpose of research (research question), unit of analysis and other practical and methodological consideration. Given that this research identifies important processes (including traditional and soft factors) that provide support for COTS software evaluation and selection, a field study of organisations with experience in CBS and COTS software evaluation was required.

Furthermore, as the research aims at providing a generic framework for COTS software evaluation and selection (i.e. STACE), evaluation of the framework was carried out in a developing country (Zambia). This is consistent with the recommendation of Fowler (1994) that in cross-border technology transfer the facilitator must be knowledgeable about both the technology and its target organisations (see section 2.3.4). Furthermore, as discussed in section 3.2.3, organisations in DCs are turning to COTS software and becoming familiar with them because of specific benefits, such as reducing software development and maintenance costs. Therefore, there is much experience in evaluation and use of COTS software in DCs. In addition, the characteristics and problems of a developing country like Zambia (see section 2.3.5) make it an excellent place for evaluating a generic framework such as STACE so as to provide more insight into the limitations of the framework.

4.3 Organisation of the research

The research process consisted of three main studies, supported by literature review activities (see figure 4-1). The purpose of the literature review was to identify the research gap in the field of COTS-based systems and information systems for developing countries. Following this literature review, the first field study aimed at eliciting and synthesising current CBS practices using a survey strategy. The outcome of the first study highlighted the problem of COTS software evaluation and selection for CBS, which resulted in a more focussed direction for the research.

Analysis of COTS software evaluation and selection frameworks reviewed a lack of addressing non-technical issues satisfactorily (see section 3.4.5). Therefore, the second field study aimed at identifying important processes (including traditional and soft factors) that support COTS software evaluation and selection was undertaken using a series of interviews in a number of organisations. The outcome of the second study was the development of a generic social-technical framework for COTS software evaluation and selection (i.e., STACE framework). The framework incorporates the often-neglected non-technical issues.

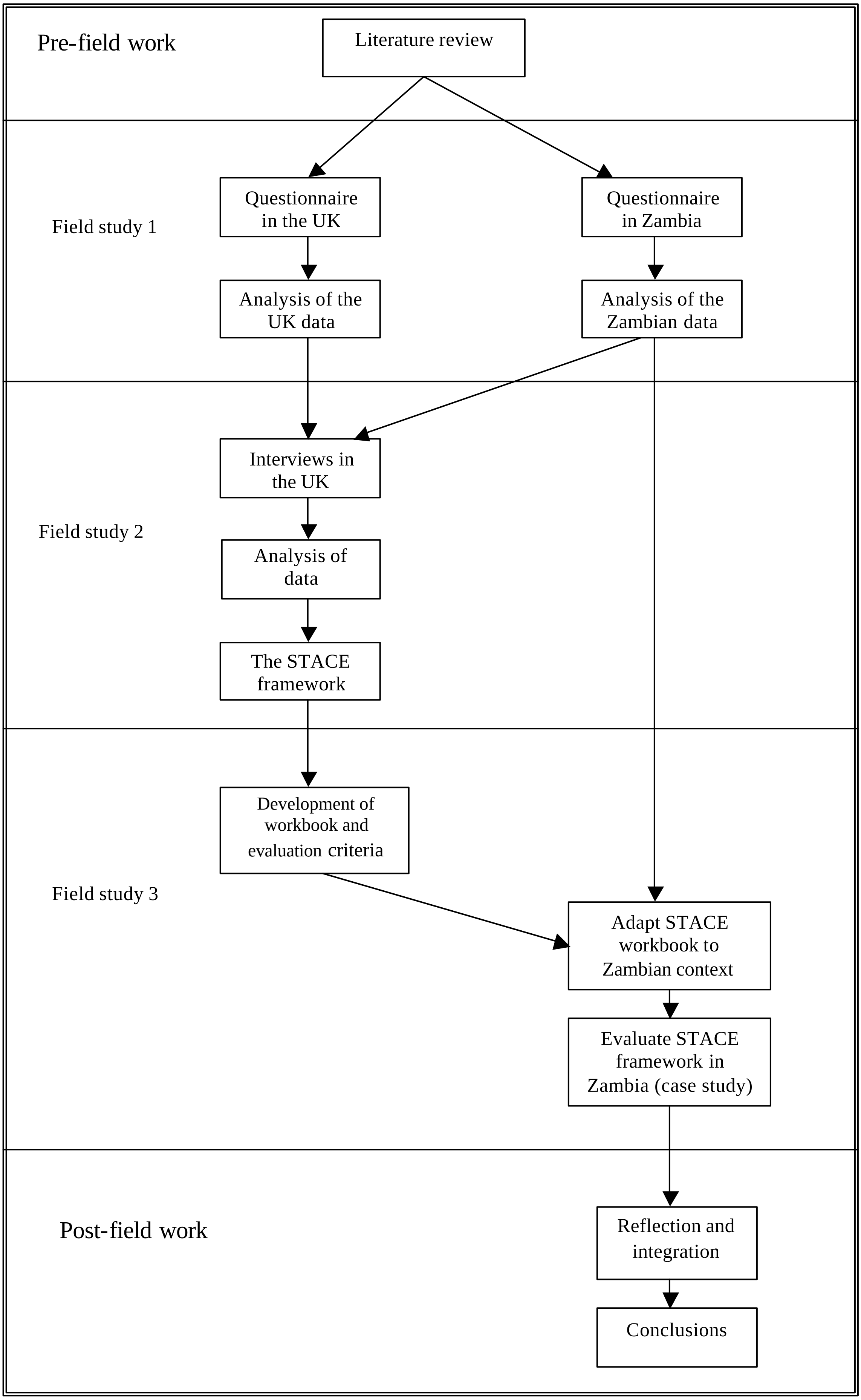


Figure 4-1. Flow diagram showing the stages of the research project

The third study, used a case study approach to evaluate the effectiveness of the STACE framework. The evaluation criteria for the STACE framework included mechanisms for assessing how STACE addresses the often-neglected non-technical issues highlighted in section 3.4.5. This was achieved by assessing the usefulness of customer participation and social-technical criteria, which are considered important features in STACE for incorporating non-technical issues. The outcome of the third study was confirmation of the validity of STACE framework for COTS software evaluation and selection supporting CBS.

Table 4-1 summarises the three studies, for each indicating the objectives, research questions, and research strategy. The data collection and analysis procedures for each study are discussed further in the following sections.

	Study 1	**Study 2**	**Study 3**
Objective	Elicit and synthesise current practices and potential benefits of CBS (also facilitate the identification of problems and solution with CBS)	Identify important factors that support COTS software selection (also develop framework for COTS software selection)	Evaluate the effectiveness of the STACE framework (also facilitated the adaptation and refinement of the framework)
Research questions	What are the current practices, process and techniques for building systems using COTS software? What are the benefits, costs and risks associated with CBS? What kinds of problems (and solutions) related to CBS development? What are the difference between a developing country (Zambia) and the UK in terms of CBS?	What are the most important processes and factors that support COTS software evaluation and selection? How can these processes and factors be classified (and how do they relate to each other)? How can the social-technical approach used to improve the COTS software selection? What kinds of problems (and solutions) have organisations experienced in evaluating COTS components for CBS?	How is the framework rated in terms of gain, interface, quality of life and task support satisfaction? How is the framework to be adapted to developing country context? What are limitations (and recommended improvements) of the framework? How are the special features and principles of the STACE framework rated in terms of usefulness?
Research strategy	Survey (self-completion questionnaire)	Theory development/ scoping study (interviews and documentation)	Case study (interviews and documentation)

Table 4-1. Organisation of the research

4.4 Eliciting current COTS-based systems practices (study 1)

The overall goal of the first study was to elicit and synthesise current practices and potential benefits of CBS (from the UK and Zambia). As discussed in section 1.4 one of the immediate objectives arising from the central research is to achieve a more comprehensive understanding of how CBS can provide support for organisations. Therefore, it is important to understand the current situation and the problems facing organisations and the potential benefits of building systems from COTS software. Eliciting current CBS practices from the UK and Zambia would also assist to draw out both the similarities and differences between a developed and developing country.

The research questions addressed during this study are provided in table 4-1. The findings of this study and their significance on this research are presented in chapter 5. The remainder of this section describes the methodology followed to achieve the overall goal of this study.

4.4.1 Research strategy

Robson (1993) points out that the purpose of the enquiry may help in selecting the research strategy, for example surveys are appropriate for descriptive studies while case studies are appropriate for exploratory work. A survey research strategy was adopted to elicit current CBS practices from the UK and Zambia.

The reasons for selecting the survey approach are; first, the purpose of the study was descriptive, i.e., aimed at describing and documenting the current practices and potential benefits of CBS from the UK and Zambia. Second, surveys are appropriate where the researcher has little control over events under study. Finally, as Yin (1994) suggests, the “what” or “how much/many” type of research questions favour survey strategy.

4.4.2 Data collection method

There are several data collection methods for a survey research strategy, for example self-administered questionnaires, interview, data archives and structured observation (Robson, 1993). Data for this study was collected through the administration of self-completion questionnaires. This method was adopted because questionnaires provide wide access to geographically dispersed samples at low cost i.e., a large population can be surveyed relatively cheaply (Nachmias and Nachmias, 1996). In addition, questionnaires provide a high degree of anonymity, which can encourage frankness when sensitive areas are involved (Robson, 1993).

However, the major criticism against questionnaires is the low response rate (Robson, 1993; Weisberg *et al*, 1996). In addition, the characteristics of non-respondents are usually not known, therefore making it difficult to know whether the sample is representative. However, there are various strategies to overcome the difficulty of securing an acceptable response rate to postal questionnaires. The strategies followed in this study include:

- Cover letter convincing the respondents to fill out the questionnaire (Nachmias and Nachmias, 1996). It should identify the sponsor of the study, explain its purpose, tell the respondents why it is important and assure them that answers will be held in strict confidence (see appendix 2).
- Order of questions, for example by ensuring that the initial questions are easy and interesting (Babbie, 1998).
- The wording of questions, clarity of wording and simplicity of design are important for securing a good response rate (Robson, 1993).
- Timing of questionnaire arrival, for example summer and holidays produce lowest response rate (Nachmias and Nachmias, 1996).
- Follow - up letter explaining the importance of the study and the value of the respondents participation (Robson, 1993). See appendix 2, for the survey reminder card used in this manner.

4.4.3 Instrument development and pilot testing

Babbie (1998) argues that the quality and trustworthiness of any research depends on validity and reliability. Instrument validity is concerned with the question of whether an instrument measures what it is supposed to measure (Easterby-Smith *et al,* 1991). Babbie (1998) discusses four types of validity: content validity, face validity, criterion-related validity and construct validity. Content validity refers to how much a measure covers the range of meanings included within the concept (Babbie, 1998). To ensure content validity, an extensive survey of relevant literature was undertaken to understand the most important elements of CBS and information systems in DCs.

Construct validity is based on the logical relationships among variables and therefore is concerned with the how they relate in accordance to theoretical expectations (Babbie, 1998). Robson (1993) indicates that face validity and criterion-related validity can be used to determine construct validity. In this study, face validity was used instead of criterion-related validity because, as Weisberg *et al* (1996) suggest, it is often difficult to obtain the external criterion against which measures can be compared.

Face validity assesses the relevance of an instrument to the characteristics of the variable it is intended to measure, which is achieved by the researcher's subjective assessment of the instrument's appropriateness (Nachmias and Nachmias, 1996). Three faculty members of the Department of Computer Science, University of York reviewed the instrument and therefore the researcher was confident that the instrument satisfied any face validity.

The questionnaire was pilot tested with two practitioners with extensive experience from both the UK and Zambia providing an opportunity for the respondents to give comments explaining their responses. The pre-testing of the questionnaire by faculty members focussed on the questionnaire validity (measuring the phenomena intended), completeness (including all relevant items), and readability (making it less likely that respondents will misinterpret the questions). As a result of the pre-testing and pilot study feedback, some questions were further modified to improve readability and remove superfluous questions. The most significant modifications were the removal of superfluous questions and the use of the 5-point scaled response questions. The final questionnaire is attached as appendix 3.

4.4.4 Sampling procedures

Robson (1993) points out the importance of sampling, for example that it is closely related to the external validity or generalisability of the findings in an enquiry; the extent to which what has been found out in a particular situation at a particular time applies more generally. It is impractical and expensive to include the entire population in a survey and researchers use sampling to select the respondents (or research units) that will be included in the study.

Weisberg *et al* (1996) distinguish between non-probability sampling procedures and probability sampling procedures. Probability or random sampling is a procedure that gives each sampling unit in the population an equal chance of being part of the sample and thus eliminates the possibility that any portion of the population being over represented or under represented in the sample. Examples of probability sampling include simple random samples, systematic samples, stratified samples, cluster samples and multi-stage samples. In non-probability sampling, there is no way of specifying the probability of each unit's inclusion in the sample and there is no assurance that every unit has some chance of being included (Nachmias and Nachmias, 1996). The inclusion of a sampling unit is at the researcher's discretion and therefore non-probability sampling can give biased results.

In this study, a systematic sampling was adopted because statistical inferences can be made about the population from responses of the sample. Furthermore, systematic sampling is more convenient compared to other probability sampling, such as simple random sampling. Systematic sampling consists of selecting every *K*th sampling unit of the population after the first sampling unit is selected at random from the total of sampling units (Nachmias and Nachmias, 1996). Weisberg *et al* (1996) warn that one of the problems with systematic sampling is that the list could contain some periodicity but this was not a problem in this study because the lists used to construct the sample were ordered alphabetically.

Considering the expected questionnaire response rate and the number of organisations with software development experience, 500 were sent to UK SMEs and 500 to UK software houses and 130 to Zambian organisations. The UK samples were constructed from the Kompass Register (1998) while the Zambian sample was constructed from the national directory of companies and organisations. The Kompass Register was used because it perceived to be representative of the UK sample and is regularly updated. The Zambia national directory of companies was used because it is considered the most up to date list of companies and organisations operating in

Zambia. The sources of data were personnel within these organisations who are responsible for specifying, procuring and developing software systems.

4.4.5 Data analysis procedures

The data collected were coded and entered in SPSS ver 9.0 (Statistical Package for Social Science) for analysis. Frequency distributions were used to categorise the demographic data variables. The *mean* values were calculated to measure the central tendency of the variables. Unlike the mode or median the mean takes into account all the values in the distribution, making it sensitive to extreme values (Nachmias and Nachmias, 1996). The *standard deviation* was used to measure the extent of dispersion (variation) from the central value, because it is more stable from sample to sample and can be used for two or more combined groups (Nachmias and Nachmias, 1996). The mean and standard deviation together were used as a standard to compare the relative importance of the variables.

The *Kruskal-Wallis H* test was initially used to investigate whether there was any significant difference between the three samples (UK SME, UK software houses and Zambian organisations). The *Kruskal-Wallis H* test can be used for three or more unrelated samples to determine the number of times a score from one of the samples is ranked higher than a score from the other sample (Cramer, 1998). If there is little difference between the sets of scores, their mean ranks should be similar. The *Kruskal-Wallis H* test is a non-parametric test and therefore the test does not depend on assumptions about the precise form of the distribution of the sampled populations. The level of significance for this study was set at 5% (i.e. p=0.05) as recommended by most researchers (Weisberg *et al*, 1996; Nachmias and Nachmias, 1996; Sapsford, 1999).

However, the *Kruskal-Wallis H* test only shows whether there is a significant difference between two or more groups but does not show which two groups differ. Therefore, *post hoc* comparisons were made using the Scheffe test to know where the difference of the groups lies. The Scheffe test is the most conservative of the *post hoc* tests in the sense that it is least likely to find significant differences between groups or making Type 1 error (accepting difference when there is no difference) (Bryman and Cramer, 1997). The Scheffe test is an F ratio in which the difference between the means of two groups is compared against an appropriately weighted within-groups mean square (Cramer, 1998). The formula for this ratio is:

$$F = \frac{(\text{one group mean - another group mean})^2}{\text{within group mean square} \times (N_1 + N_2)/(N_1 \times N_2)}$$

Where N_1 and N_2 is the number of cases in the two groups being compared. The results of a Scheffe test also indicates the relative size of the difference between two groups to a specified order of significance, such as 0.05 and the confidence interval within which a difference can be found.

The next section describes the research method adopted for the second study aimed at identifying important processes and factors that support COTS software selection.

4.5 Identifying factors for COTS selection (study 2)

The overall goal of the second study was to identify important processes/ factors (including traditional and soft factors) that support COTS software selection for CBS from the UK (see section 1.4). The first study brought out a number of problems associated with building systems from COTS software, especially the problem of COTS software evaluation and selection. COTS software evaluation and selection is problematic because of a number of reasons such as rapid changes in marketplace, misuse of data consolidation methods (see section 3.3.1). Moreover, existing framework for COTS software evaluation and selection tend to focus on technical neglecting the equally important non-technical issues (see section 3.4.5).

Therefore, identifying important processes and factors would facilitate the development of a framework for COTS software evaluation and selection that addresses these problems and incorporates the often-neglected non-technical issues. The research questions investigated in the second study are provided in table 4-1. (See chapter 6 for details of the findings of this study and their significance on this research). Table 4-2 provides an overview of the steps taken and rationale for each step.

PHASE	ACTIVITY	RATIONALE
RESEARCH DESIGN		
Step 1: Develop interview protocol	Review of technical literature	Increases reliability
	Definition of research questions	Defines research focus
	Definition of construct and field procedures	Sharpens construct validity of the research
Step 2: Select theoretically relevant cases	Theoretical sampling	Confirms, extends, and sharpens theoretical framework.
DATA COLLECTION		
Step 3: Use rigorous data collection procedures	Use interviews and documentation to collect data	Strengthens field study through triangulation, increases construct validity
DATA ANALYSIS		
Step 4: Field study database	Use ATLAS/ti qualitative software tool for data analysis	Establishing chain of evidence and increases construct validity and use of software tool enhances reliability
Step 5: Categorising	Open coding	Develops concepts, categories and properties
	Axial coding	Develops connections between a category and its sub-categories
Step 6: Data display and tabulating	Event list matrix	Present data in order to draw valid conclusions
	Effects matrix	Present data in order to draw valid conclusions
Step 7: Explanation – building	Explanatory effects matrix	Enhances internal validity
	Selective coding	Integrate categories to build theoretical framework
	Memoing	Used to capture ideas, views and intuitions
Step 8: Modelling	Event-state network	Present data in order to draw valid conclusions
	Causal network	Presents variables and relationship among them
FIELD STUDY CLOSURE		
Step 9: Review findings	Review of individual and cross-company reports by key informants	Increases construct validity
Step 10: Comparison with literature	Comparing with literature conflicting findings	Improves construct definitions and therefore internal validity
	Comparing with literature supporting findings	Also improves external validity by establishing domain in which the study's findings can be generalised
Step 11: Theoretical saturation	Achieve theoretical saturation	Ends process when marginal improvement becomes small

Table 4-2. Field study research process (adapted from Pandit, 1996)

4.5.1 Research strategy

A field study approach comprising of a set of interviews was used to identify the important processes/factors that support COTS component selection in CBS. The interviews were conducted in a range of organisations with vast experience in CBS and COTS software selection processes. The advantage with this approach is that it allows cross-organisation analysis and comparison, and the investigation of a particular phenomenon in diverse settings. Therefore, the more organisations included in the study, the more evidence would accumulate, the more variations would be found and the greater density would be achieved. Multiple comparisons between purposefully selected organisations are crucial for a qualitative study to identify patterns and to develop theoretical categories.

4.5.2 Data collection procedures

The dominant data collection method adopted in this field study were interviews. The primary purpose of the interview is to understand the meanings interviewees attach to issues and situations in context that are not structured in advance by the researcher's assumptions (Easterby-Smith *et al*, 1991). It is an opportunity for the researcher to probe deeply to uncover new clues, open up new dimensions of a problem and to secure vivid, accurate inclusive accounts that are based on personal experience.

The major advantage with the interview approach is its flexibility and adaptability (Robson, 1993). For example, interviews offer the possibility of modifying one's line of enquiry, following up interesting responses and investigating underlying motives. However, Easterby-Smith *et al* (1991) highlight a number of issues that must be considered in order for interviews to be successful. These include the problem of obtaining trust, the need to understand social interactions between interviewer and interviewee, how much structure to put in the interview, and the problem of interviewing skills and bias. A number of procedures to deal with these problems and increase the quality of data collection operations were followed in this study (Easterby-Smith *et al*, 1991; Robson, 1993; Nachmias and Nachmias, 1996):

- The interview was conducted in an informal and relaxed atmosphere, and attempts were made to avoid creating the impression that what was occurring was a cross-examination or a quiz. This was achieved by putting the questions in a straightforward, clear and non-threatening way.
- Focused or semi-structured interviews were adopted because they provide insight into the phenomena being studied as well as being amenable to interpretation. Semi-structured interviews provide respondents with considerable liberty to express their views.

- The technique of probing was used to motivate the respondent to elaborate on or clarify an answer or to explain the reasons behind the answer and focus the conversation on the specific topic of the interview.
- Long questions, double-barrelled questions, leading questions and biased questions were avoided.
- Question that were misinterpreted or misunderstood were repeated or clarified, avoiding comments that induce particular answer (e.g., "don't you think ….").

The interviews lasted from one hour to three hours and were tape recorded and then transcribed. Interviews were supplemented with documentary evidence. Documentary evidence involves collection, evaluation and analysis of relevant documents from the research sites. The disadvantage with collecting data from multiple sources is that it is more expensive than when data were only collected from a single source, for example the cost of time and effort to collect the documents. However, the advantage of using multiple sources of evidence is the development of converging lines of inquiry, a process of triangulation (Yin, 1994). The findings of a study is likely to be much more convincing and accurate if it is based on several different sources of information, following a corroboratory mode.

4.5.3 Pilot study and protocol development

An interview protocol was developed and used to provide guidance in data collection procedures. The preparation of the protocol involved extensive review of technical literature, definition of research questions and definition of construct and field procedures. The definition of research questions assisted in refining the research focus, while the definition of construct and field procedures helped to sharpen the construct validity of the research.

Two faculty members of the Department of Computer Science, University of York initially reviewed the interview protocol. Then the protocol was pilot tested with one organisation with experience in CBS and COTS software evaluation and selection. This organisation was also selected because of proximity and availability of informants. The review by faculty members and pilot study assisted in validating the interview protocol and field procedures. In addition, piloting helped to clarify data analysis procedures. The final interview protocol is attached as appendix 4.

4.5.4 Data analysis procedures

The data collected (i.e., write-up notes, interview transcripts, session summary sheet and documentation) was entered into ATLAS/ti™ software tool for analysis, which acted as a field study database. This procedure of maintaining a chain of evidence is another strategy that increases the reliability of the information in a field study; it allows an external observer to follow the derivation of any evidence from initial research questions to ultimate study conclusions.

The general mode of analysis adopted for the second study is explanation building, which involved the following steps: categorising (developing concepts through coding), data display or tabulating, explanation-building and modelling (building causal networks).

a. Categorising

Categorising themes in a field study is the first step in data reduction. This was done by the development of coding categories, in which a code is a symbol applied to a group of words to classify/categorise them, as they relate to the research question, concepts and themes (Robson, 1993). Codes are retrieval and organising devices that allow the researcher to find and then collect together all instances of a particular kind. Using ATLAS/ti software, noteworthy segments relevant to one idea were retrieved and compared to one another until patterns began emerging. Miles and Huberman (1994) distinguish between first level coding and pattern coding, which is similar to *open coding* and *axial coding* in grounded theory (Strauss and Corbin, 1990). Figure 4-2 illustrates the coding scheme for assessment (evaluation).

b. Data display or tabulating

Robson (1993) points out that data display methods are used to present data in such a manner that valid conclusions can be drawn. Matrices are the most commonly utilised data display methods and are used to describe the flow of events and processes since time is a crucial aspect of the study (Miles and Huberman, 1994). The adopted matrices for data display in this study were *event list* and *effects* matrix.

```
HU:  Case 8
File:  [C:\KUNDAD~1\IDENTI~1\ORG8]
Edited by: Super
Date/Time: 1998/12/26 - 10:38:14
---------------------------------------

Data analysis techniques   <is part of>  Assessment (evaluation)
    Analytic Hierarchy Process (AHP) <is a> Data analysis techniques
    Feature analysis  <is part of>  Data analysis techniques
    Weighted Sum Method (WSM)  <is a>  Data analysis techniques
Data collection techniques   <is part of>  Assessment (evaluation)
    Audit development process  <is a>  Data collection techniques
    Benchmark testing  <is a>  Data collection techniques
    Demonstrations  <is a>  Data collection techniques
    Experimentation  <is a>  Data collection techniques
    Pilot studies  <is a>  Data collection techniques
    Qualification of COTS  <is a>  Data collection techniques
    Software tests  <is a>  Data collection techniques
    Study documentation  <is a>  Data collection techniques
    User community, evaluation  <is a>  Data collection techniques
Evaluation strategy  <is part of>  Assessment (evaluation)
    Adhoc (non-systematic)  <is a>  Evaluation strategy
    Keystone identification  <is a>  Evaluation strategy
    Other approaches e.g. PRINCE  <is part of> Evaluation strategy
    Progressive strategy  <is a>  Evaluation strategy
    Puzzle assembly  <is a>  Evaluation strategy
```

Open coding

Axial coding

Figure 4-2. An example of code hierarchy for assessment (evaluation)

An event listing is a matrix that arranges a series of concrete events by chronological time periods, sorting them in several categories (Miles and Huberman, 1994). To produce an event listing, the quotations that matched the code *events* were extracted from ATLAS/ti, as a report. This was also cross-referenced to other codes or themes. The events are then re-arranged in chronological order and presented in the *event list* matrix. The *effects* matrix are discussed in the next section.

c. Explanation - building

This stage involved exploring and explaining the data. To "explain" a phenomenon is to stipulate a set of causal links about it (Yin, 1994). In order to support good explanations the explanatory effects matrix were generated. An explanatory effects matrix displays data on one or two outcomes (i.e. on dependent variables) of a process, as the study requires (Miles and Huberman, 1994).

Figure 4-3 shows an example of the effects matrix that was generated by searching for the codes of the effects of requirements definition on evaluation (assessments). The quotations were then categorised according to confirmatory or contradictory evidence.

An explanation then was formulated and conclusions were confirmed (i.e. tested or verified) by focusing on negative evidence or checking out rival explanations. Miles and Huberman, (1994) suggest a number of ways in which conclusions can be drawn from the effects matrix, for example by making conceptual/theoretical coherence, building a logical chain of evidence and using a tactic of counting or comparing and contrasting. Applying the logical chain of evidence enhances construct validity, while explanation building and effects matrix increases internal validity.

Variable 1: Requirements definition
Variable 2: Evaluation (assessment) process
Effect: The evaluation (assessment) process depends on the requirements definition.

Confirmatory evidence
In terms of best practices, it is recommended that you let users draw up specifications of what they want before embarking on purchasing COTS packages. P 1: tJames.txt - 1:22 (79:82)

The process of evaluation, you begin with high level criteria (although we would have written requirements). You look at the documentation and this process sometimes is a nightmare because manuals are not available. You arrange training course or vendor demonstration in order to understand the package. P 2: tDavid.txt - 1:39 (53:58)

Contradictory evidence
In component selection, you have first to find out who has the components that meet your need, you may find the component out there that meet half your needs and the other half does not or too much your needs. You have to make a decision whether to get a component that gives more than you actually need and pay for that or get component that delivers slightly less than you need but allow to extend it. P 3: tJohn.txt - 2:8 (99:103).

In COTS software selection, you should search for alternative COTS packages before procuring and let the supplier do some presentations. It is also important to understand the licensing arrangements before procurement. P 1: tJames.txt - 1:23 (83:85)

Explanations
E1: The importance and effect of the requirements definition on the evaluation (assessment) process is supported by the quotations above. The respondents were arguing that it important to define the requirements prior to evaluation (i.e., COTS software selection must be driven by requirements).
E2: The contradictory evidence suggests that COTS software selection process must be driven by what is available in the marketplace. However, this is not contradictory because to realise the benefits from COTS software, the requirements must be defined according to what is available from the marketplace.

Figure 4-3. Effects Matrix of requirements definition on the evaluation (assessment)

The relationships in the effect matrices were entered as network links and explanations were stored as memos in the ATLAS/ti software. The process of generating effects matrices can be seen as similar to the selective coding in grounded theory (Strauss and Corbin, 1990).

d. Modelling

This stage involves linking variables or factors together into models or networks. Two types of models are used in this research, namely event-state network and causal network. These are visual representation of both the axial and selective coding of the grounded theory method. Event-state network is a simple way to visually display the difference between events and states as shown in the event list. This is derived from the event list of the data-display phase (section 4.5.4b). In this study, the event-state network modelling approach was used to identify and understand the chronology of the processes in COTS software evaluation and selection.

A causal model or network is a display of the most important independent and dependent variables in a field study (shown in boxes) and of the relationships between them, shown as arrows (Miles and Huberman, 1994). This was derived from the explanation effects matrix and selective coding. ATLAS/ti produces the network from the relationships arising from axial coding and the selective coding process. Figure 4-4 shows an example of a representative causal network.

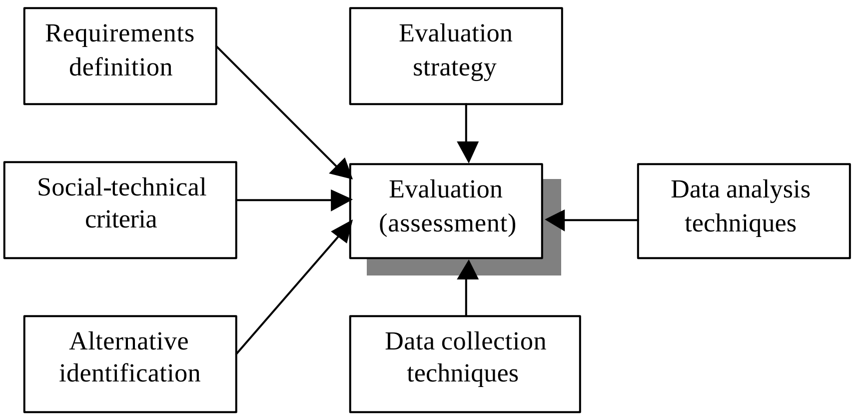


Figure 4-4. A causal network of evaluation (assessment)

In this study, data analysis followed an iterative process starting with some theoretical propositions based on data from the first organisation visited and then systematically comparing them with the findings of the second organisation. The emergent propositions were then refined and compared to third findings and the process was repeated until the eighth organisation was included. The resulting theoretical propositions from eighth organisation were again compared to the first organisation and the process was repeated again. The central idea of this iterative process was to search for patterns (Pare and Elam, 1997). The outcome of this theory-building or creative conceptualisation led to the learning of important lessons.

4.5.5 Review of findings and field study closure

In this study, the findings from organisations visited were compiled into reports and sent for comments to key informants and other relevant personnel in the organisations before proceeding to the next interview site. The purpose of the review process was to corroborate evidence and to increase construct validity. This also provides an opportunity to begin fresh dialogue about future research collaboration. In this study, the review process also produced further evidence for the researcher as new ideas and materials emerged that were not available during the data collection phase. For example, one informant after reviewing the draft report provided documentation that clarified some of the issues presented earlier during the interview.

Ideally, the selection of additional material should stop when theoretical saturation of each category is reached (Eisenhardt, 1989). That is until no new or relevant data seem to emerge regarding a category and the category development is dense, insofar as all the paradigm elements are accounted for, along with variation and process (Strauss and Corbin, 1990). Theoretical saturation is also achieved when the relationships between categories are well established and validated. Theoretical saturation is important because the theory developed would be conceptually inadequate if saturation is not reached. Although only eight organisations participated in this study, the iterative approach of theoretical sampling used ensured variations and greater density in data, which led to an increase in the generality of the findings.

The next section describes the research method adopted for the third study.

4.6 Evaluating STACE framework (Study 3)

The objective of the third study was to evaluate the effectiveness of the STACE framework to support COTS software evaluation and selection for CBS. As a result of the second study, a generic social-technical framework for COTS software evaluation and selection (i.e., STACE) was developed. Therefore, the evaluation exercise would confirm the validity of the STACE framework, showing how a social-technical approach to COTS software evaluation and selection can support CBS (see section 1.4). Furthermore, evaluation would facilitate improvements and refinement to the STACE framework. The research questions addressed during this study are provided in table 4-1.

4.6.1 Research strategy

A case study research strategy was adopted for this study. A case study is preferred when the investigator cannot manipulate or control the relevant behavioural events. Darke *et al* (1998) points out that case studies are well suited to understanding interactions between information technology related innovations and organisational contexts. The adopted research design for this study was a multiple-case study approach because it allows for cross-case analysis and comparison.

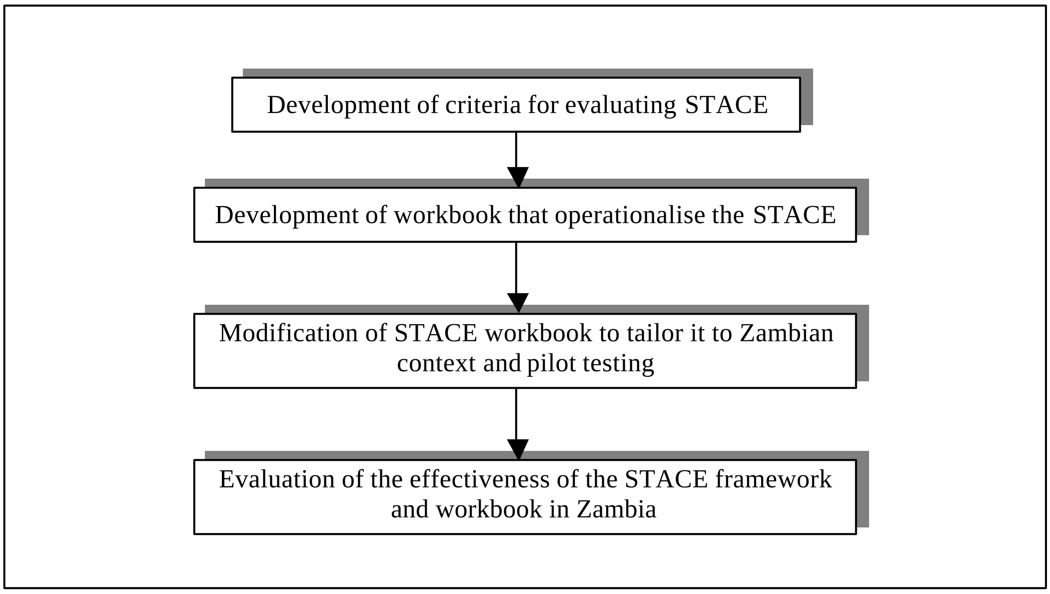


Figure 4-5. Evaluating the STACE framework

The evaluation process involved the following activities (see figure 4-5): development of the evaluation criteria (see section 4.6.2); development of workbook that operationalise the STACE and pilot study (see section 4.6.3); modification of workbook to Zambian context (see sections 4.6.3 and 7.3); and 4) evaluation with nine organisations in Zambia. The participating organisations were provided with the materials on the STACE framework and the workbook in advance and were requested to apply the framework to a "real" COTS software selection exercise. This was then followed by data collection aimed at evaluating the effectiveness of the framework.

4.6.2 Development criteria for evaluating the STACE framework

An information system is said to be effective if it produces the desired result for which it was developed. Researchers have developed a number of indicators for measuring system success. Kitchenham (1998) in evaluating the DESMET method used three major measures of success: basic evaluation, use evaluation and gain evaluation. Basic evaluation is concerned with quality of the component documentation for example completeness, readability and understandability of the component description. Use validation is concerned with the quality of the component, for example is the component easy to implement and "helpful". Gain validation is concerned with the benefits delivered by the component, for example whether the component is cost-effective and supports decision making.

Garrity and Sanders (1998) argue that, when measuring the success of an information system, it is important to include the organisational and social-technical viewpoints. Based on theoretical and empirical data they classified measures of success into task support, quality of life support satisfaction, interface satisfaction and decision support satisfaction. In this study, the criteria for evaluating STACE framework is based on Kitchenham (1998) and the work of Garrity and Sanders (1998), which together provide a comprehensive taxonomy of the important success variables including the social-technical viewpoints (see table 4-3).

The evaluation criteria comprise gain validation, interface satisfaction, quality of life and task support satisfaction. Gain validation measures whether the framework would be beneficial, for example perceived usefulness (Kitchenham, 1998). Interface satisfaction focuses on the characteristics of the interface in terms of presentation, format and processing efficiency (Garrity and Sanders, 1998). Quality of life satisfaction measures an individual's affective response to the method in terms of how the system affects quality of work-life and job satisfaction (Garrity and Sanders, 1998). Task support satisfaction measures whether the framework would achieve its intended objectives and satisfies evaluators (Kitchenham, 1998).

Evaluation criteria	Research variables
Gain satisfaction (Kitchenham, 1998).	Perceived usefulness Decision support satisfaction Comparison with other guidance - better Cost – effectiveness Clarity - clear and illuminate the process Appropriateness for task
Interface satisfaction (Garrity and Sanders, 1998).	Perceived easy of use Appropriate for audience Organisation - well organised Internally consistent Presentation - readable and useful format
Quality of life satisfaction (Garrity and Sanders, 1998)	User feeling of participation
Task support satisfaction (Kitchenham *et al* 1997; Garrity and Sanders, 1998).	Ability to produce expected results Ability to produce relevant results Ability to produce usable results Completeness - adequate or sufficient Ease of implementation Understandability - simple to understand

Table 4-3. Research variables for evaluating the STACE framework

A number of special features and principles of the STACE framework were identified in study 2 (see section 6.5). These include the customer participation, social-technical criteria, Internet to identify COTS software and AHP. The usefulness of these features was evaluated to facilitate improvements to the framework. The respondents were also requested to provide problems (and recommendations) experienced during the evaluation exercise or potential problems that they would experience when they adopt the STACE framework in their organisations. The development of the evaluation criteria was the basis for developing a data collection protocol, which provided guidance on the field procedures for data collection (see appendix 7).

4.6.3 Development of STACE workbook and pilot study

The STACE workbook was developed by the researcher to operationalise the STACE framework and was used as a guide by participating organisations for evaluating and selecting COTS software for CBS. The objective of developing a workbook was to promote uniformity in the work of participating organisations. The workbook explicitly describes each stage of the STACE framework and provides an example to illustrate the use of the STACE framework. The advantage of encouraging all those participating to adopt the same STACE workbook is that it reduces the threats to internal and external validity of the research findings.

As already discussed (see section 2.3.4), many authors argue that techniques and methods developed with a different socio-cultural context should be adapted when used in DCs (Bjorn-Andersen, 1990; Janczewski, 1992; Fowler, 1994). Therefore, the first version of the STACE workbook was adapted to the Zambian context prior to evaluation in Zambia. The STACE workbook was modified based on literature on DCs and the findings of the first study aimed at eliciting current CBS practices from Zambia (see section 7.3).

Two faculty members of the Department of Computer Science, University of York reviewed the workbook. The purpose of the review was to improve the face validity of the workbook. Furthermore, the workbook was piloted tested with one organisation in Zambia. The pilot case organisation was selected because it was perceived to provide rich insight into the potential problems of the STACE framework. The key informants in this organisation had vast experience in COTS software evaluation and selection. In addition, the researcher had easy access to this organisation. The piloting of the workbook in Zambia provided an opportunity to improve the workbook in terms of completeness and readability. It also assisted in refining the evaluation procedures and case study protocol. As a result of pilot study and review by faculty members, a second version of the workbook was produced (see appendix 6).

4.6.4 Data collection and analysis

Interviews and documentary evidence were used to collect data for this study. A data collection protocol was developed and used to guide field procedures and data collection (see appendix 7). In order to avoid interview bias efforts were made to ask open-ended questions in a neutral way and then respondents were asked to explain their responses. The responses that were not supported with an explanation were disregarded in the analysis. As indicated before in section 4.5.2, multiple sources of evidence provide a converging line of inquiry and enhance construct validity. The advantages and disadvantage of using this data collection approach were also presented in section 4.5.2.

The data collected was entered into the ATLAS/ti software tool for analysis, which was also used as case study database. The adopted mode of analysis for this study is similar to explanation building as discussed in section 4.5.3. This involved coding the data around the categories initially developed in section 4.6.2 (i.e., criteria for evaluating STACE's effectiveness). Furthermore, it involved developing coding categories around the limitations of the framework brought out by the respondents. Then explanations were formulated and conclusions were confirmed (i.e., tested or verified) by focusing on negative evidence or checking out rival explanations.

The preliminary findings were compiled into draft individual case study reports and reviewed by key informants to increase construct validity. The individual case study findings were systematically compared in an iterative manner to search for cross-case patterns.

4.7 Validity and reliability of research findings

The validity and reliability of the research establish the quality of any empirical research. Robson (1993) points out that validity is concerned with establishing the value and trustworthiness of an enquiry while reliability is essentially a quality control measure. Precautionary measures were taken in this research to identify all possible sources of error and avoid them. Furthermore, great care was taken to improve the riguor of the three studies, and enhance construct validity, internal validity, external validity and reliability. The validity and reliability of the first study was discussed in section 4.4.3.

In the second and third study three strategies were used to increase construct validity. Firstly, the studies relied on interviews and company documentation to assess the phenomena. Miles and Huberman (1994) argue that multiple sources of evidence (triangulation of data) can be used to counteract biases in the researcher's collection and analysis of data, thus increase construct validity. Secondly, by establishing a chain of evidence in such a manner that an external observer is able to follow the derivation of any evidence from initial research questions to the conclusions. This was achieved by developing data collection protocol based on literature and using the software tool. Lastly, the draft reports were produced and reviewed by key informants. The review process by key informants facilitated the validation of operational concepts identified in the studies.

Internal validity is concerned with establishing a causal relationship, whereby certain conditions are shown to lead to other conditions, as distinguished from spurious relationships (Yin, 1994). Explanation building was used to analyse qualitative research data to enhance internal validity (see sections 4.5.4 and 4.6.4). A continuous effort was made to identify other moderating factors and to check for negative evidence or rival explanations to the conceptual model, until data saturation was reached. In order to reduce the researcher's bias the draft study reports were reviewed by key informants.

External validity is concerned with the generalisability of the findings, a study with low external validity offer a poor basis for generalisation (Lee, 1989). Pandit (1996) points out that comparing the findings with literature can improve external validity by establishing domains in which study's findings can be generalised. In this study, the findings and conclusion were compared with literature. Furthermore, Kelle (1995) suggests that an increase in the sample size allied to the use of software tool add greater breadth to the scope of the analysis while maintaining the depth of interpretation which can be regarded as the hallmark of qualitative analysis technique.

Reliability is concerned with demonstrating that the operations of a study such as the data collection procedures can be repeated, with the same results. To improve the reliability of the study, a data collection protocol was developed and used to guide the studies (see section 4.5.2). Applying the protocol assisted in ensuring that the same questions and procedures were consistently followed when collecting data. Furthermore, a software tool was used to support the data analysis procedures. Kelle (1995) argues that the use of a software tool increases the trustworthiness of qualitative findings considerably because these facilities can ensure that the hypotheses developed are grounded in the data and not based on single and highly untypical incidents.

4.8 Summary

This chapter has presented the research methods used in different phases of the research. The chapter first described the research design and framework that outlined the research purpose, unit of analysis and research location. The chapter then presented the organisation of the research. The purpose of the organisation of the research was to provide an overview of the three main studies and how they relate to each other.

The chapter then described the research strategy, data collection and analysis procedures for each of the three main studies. The first study was aimed at eliciting and synthesising current practices and potential benefits of CBS in the UK and Zambia. The study provided an understanding of CBS practices and was responsible for directing this research. A survey approach was adopted for the first study. This involved instrument development, selection of sample, administration of a self-completion questionnaire to the sample organisations and statistical analysis.

The second study was aimed at identifying important processes (including traditional and soft factors) that support COTS software components evaluation and selection for CBS from the UK. A field study approach comprising a series of interviews was adopted for the second study. The study led to development of a framework COTS software evaluation and selection that incorporates the often-neglected non-technical issues (i.e., the STACE framework). The third study evaluated the effectiveness of the STACE framework in Zambia. A case study approach was adopted for third study. Research data were collected using semi-structured interviews and documentation. Data analysis involved development concepts through coding and explanation building. The next chapter presents the results of the first study aimed at eliciting and synthesising current CBS practices from the UK and Zambia.

5 Eliciting current COTS-Based Systems practices (study 1)

This chapter presents the results of the first study aimed at eliciting and synthesising current CBS practices from the UK and Zambia. Data was collected through self-completion questionnaires in the sample organisations. Descriptive and non-parametric tests were used to obtain an understanding of the current practices, techniques and problems related to building systems from COTS software. The significance of results on the overall research is also discussed in this chapter.

5.1 Introduction

The overall goal of the first study was to elicit and synthesise current practices and benefits of CBS from the UK and Zambia (see section 1.4). In this thesis current CBS practices means established procedures, methods and approaches adopted by organisations in building systems from COTS software. This represents, in some way, the lessons they have learned about how best to build systems from COTS software.

It is important to elicit current CBS practices in order to obtain a better understanding of the current situation, problems (and solutions) organisations have experienced in relation to CBS. Drawing out both the similarities and differences between the UK and Zambia would facilitate identification of problems associated with building systems from COTS software. The following were the research questions investigated during this study (see table 4.1):

- What are the current practices and techniques for building systems using COTS software?
- What are the benefits, costs and risks associated with building systems from COTS software?
- What kinds of problems (and solutions) have organisations experienced from the past in relation to building systems from COTS software?
- What are the differences between the UK and Zambia regarding building systems from COTS software?

The outcome of this study were used to focus the overall research and a better understanding of how CBS can provide support for organisations to develop and implement more effective information systems. This chapter reports the results of this field study and discusses the significance of these findings for the overall research. The next section reviews the research method used to conduct this study.

5.2 Research method

The research method is more fully discussed in section 4.4. A survey approach was adopted because the purpose of this study was descriptive (see section 4.4.1), therefore, a self-completion questionnaire was developed based on the literature on social-technical approaches to information systems, information systems in DCs and CBS. The questionnaire was sent to the sample organisations and respondents where to respond within 4 weeks time period. Systematic sampling, a probability sampling procedure, was adopted so that statistical inferences can be made about the population from responses to the sample. The sample was drawn from UK software houses, UK SMEs and Zambian organisations with the potential of developing software systems using CBS approach. At the end of four weeks, the low response rate was supplemented with a reminder card, to increase the number of responses.

Statistical analysis was used to analyse the results from this survey. The *mean* and *standard deviation* were used as a standard to compare and determine the relative importance of the variables. The Kruskal-Wallis H test was used to investigate whether there was any significant difference between the three samples (UK SME, UK software houses and Zambian organisations). However, the Kruskal-Wallis H test only shows whether there is a significant difference between two or more groups but does not show which any two groups differ. Therefore, *post hoc* comparisons were made using the Scheffe test to assess the differences between the UK and Zambian samples.

5.3 Survey response rate

A survey package containing a covering letter and the survey instrument (see appendices 2 and 3) were sent to 500 UK SMEs, 500 software houses and 130 Zambian organisation. Table 5-1 shows the total number of questionnaires distributed and the response rate. The returned questionnaires are those that were returned because of wrong addresses and were excluded from data analysis. The rejected ones are those respondents who responded to the questionnaire, but either did not complete most of the questions from the questionnaire or indicated that it was their policy not to participate in surveys and therefore did not complete the questionnaire.

Description	**SME (UK)**	**Software houses (UK)**	**Zambian Organisations**
Questionnaires Sent	500	500	130
Returned	5	5	0
Usable responses	24	49	13
Rejected	2	2	0
Sub-total responses	26	51	13
Reminders	218	331	50
Usable responses	8	8	7
Rejected	2	0	0
Sub-total responses	10	8	7
TOTAL RESPONSE	**36**	**59**	**20**
Response Rate (%)	**7.2**	**11.8**	**15.4**

Table 5-1. Questionnaire response rate

The next section presents the analysis of the demographic data collected in this study.

5.4 Demographic data

Frequency distributions were used to categorise demographic data. The demographic data is presented in terms of respondent's characteristics (job function and years of experience) and company information (primary business and annual turnover).

5.4.1 Company details

The respondents were asked to indicate the primary business of the organisation and their annual turnover. Table 5-2 indicates that the primary business of the majority of the UK SMEs in this study were manufacturing, followed by construction and engineering. The primary business of the majority of the UK software houses and Zambian organisations were to provide IT services. The majority of the organisations from the UK (SMEs and software houses) indicated that their annual turnover ranges between £5million to £35million (see table 5-2). The majority of those that provided their annual turnover in Zambia indicated a turnover of either between £5million to £35million or below £1million. It was interesting to note that none of the organisations from Zambia indicated a turnover of greater than £500 million.

Theme	Factors investigated	SME (UK) (%)	UK soft house (%)	Zambia (%)
Company primary business	Banking/Finance	0	2	15
	Manufacturing	41	14	5
	Construction/Engineering	28	9	10
	Retail/Wholesale	16	2	15
	IT services	3	54	25
	Government	3	4	10
	Other	9	16	20
Company Annual turnover	Below £1million	16	11	25
	£1 - £5million	34	19	5
	£5 - £35million	47	28	25
	£35 - £100million	0	11	10
	£100 - £500million	0	11	5
	£500 - £1billion	0	2	0
	Above £1billion	0	16	0
	Don't know	3	4	30

Table 5-2. Company demographic data

5.4.2 Respondents

The respondents were asked to indicate their main job function and years of experience. The majority of the respondents in all the samples classified their job function as management (see table 5-3). The majority of the respondents in the UK SMEs and the Zambian organisation indicated that their work experience ranged from 5 to 10 years, while the majority of the UK software houses indicated that work experience ranged from 10 to 15 years. Interestingly none of the respondents from UK SMEs and the Zambian organisation indicated that they had more than 20 years of experience, while 12% of the respondents from the UK software houses indicated they had more than 20 years of work experience.

Theme	Factors investigated	SME (UK) (%)	UK soft houses (%)	Zambia (%)
Job function	Management	53	56	40
	Systems analysis	3	4	25
	Application Programming	3	0	5
	Academic or Research	3	2	20
	Other	38	39	10
Work experience (years)	Below 5 years	19	12	10
	5-10	22	11	35
	10-15	6	30	25
	15-20	6	7	5
	Above 20 years	0	12	0
	Not specified	47	28	25

Table 5-3. Respondents characteristics

The next section presents the results of descriptive and non-parametric tests used in the study to investigate the current practices and CBS benefits.

5.5 Survey findings

The respondents were asked to rate their strength of agreement to some factors related to building systems from COTS software. The questionnaire consisted of scaled-response from 1 to 5 such that 1 = *strongly disagree* and 5 = *strongly agree*. The *mean* and *standard deviation (S.D.)* were calculated and then used as standard to compare the relative importance of the variables. The Kruskal-Wallis H test was calculated and initially used to investigate whether there was any significant difference between the three samples (UK SME, UK software houses and Zambian organisations). Then *post hoc* comparisons were made using Scheffe test to assess the differences between Zambian organisation and the UK regarding building systems from COTS software.

5.5.1 Overview of development process

This section presents the survey findings related to the main constraints to developing software systems, benefits and risks of building systems using COTS software.

a. Main constraints or obstacles to developing software systems

In the literature, researchers have pointed out a number of constraints or obstacles to developing software systems. For example, Okot-uma (1992) and Bhatnagar (1992a) argue that a shortage of skilled human resources and lack of financial resources are major obstacles impinging on the development of software systems in DCs. Table 5-4 shows the ratings of the most significant constraints or obstacles to developing software systems, according to the respondents.

Factors investigated	**SME (UK)**		**UK soft houses**		**Zambia organ.**	
	Mean	**S.D.**	**Mean**	**S.D.**	**Mean**	**S.D.**
Lack of financial resources	2.74	1.52	2.84	1.09	2.68	1.29
Lack of adequate trained human resources	3.60	1.19	3.60	0.76	3.95	1.13
Lack of time	3.40	1.23	3.86	0.91	2.83	1.20
Lack of institutional support	3.25	1.37	2.60	0.98	3.17	1.50
High development costs	3.30	1.42	3.10	0.77	2.88	1.27
Political issues	2.53	1.65	2.75	1.28	2.06	1.43
External factors	2.32	1.49	2.24	1.19	2.06	1.20

Table 5-4. Main constraints or obstacles to developing software systems

The results show that, on average, the UK SMEs and Zambian respondents consider that the main constraints, or obstacles to developing software systems, is the lack of adequately trained human resources. However, the UK software houses indicated that the major problem with them is a lack of time because they are busy with lots of work and projects. Kontio (1996) also identified this problem, that organisations are under pressure to perform and therefore do not have enough time to use the most appropriate CBS methods. The relatively low standard deviations in the ratings by the UK software houses indicate that there is considerable homogeneity in opinion among the respondents about the constraints or obstacles to developing software systems.

b. Benefits and risks of building systems using COTS software

Table 5-5 indicates that, on average, the respondents from the three samples consider that the major benefit of CBS is reducing software development costs. This is consistent with literature findings that building systems from COTS software has the potential to reduce software development costs (Clements, 1996; Haines *et al*, 1997). The respondents from the UK indicated that the next most important benefit of building systems from COTS software was reducing maintenance costs. However, Zambian organisations rated improving reuse across projects as the second most important benefit of CBS.

Theme	Factors investigated	SME (UK)		UK soft houses		Zambia	
		Mean	S.D.	Mean	S.D.	Mean	S.D.
Benefits	Reduces development cost	4.38	0.62	3.95	0.96	4.18	0.95
	Reduces maintenance cost	3.63	0.81	3.64	1.14	3.89	1.02
	Improves reuse	3.19	0.98	3.38	0.96	4.06	0.75
	Obsolescence management	3.19	0.75	2.88	0.82	3.18	0.95
	Competitive market	3.38	1.02	3.23	0.97	4.00	0.91
Risks	Lack of guidelines	3.06	1.29	2.83	0.96	2.65	1.27
	Technical capability	3.81	1.17	3.86	0.84	3.47	1.42
	Periodic releases of COTS	2.94	1.06	3.60	0.99	3.94	1.25
	Loss of schedule control	2.81	1.05	2.98	0.95	2.81	1.17
	Legal implications	3.00	1.37	2.93	1.05	2.47	1.18
	Product mismatches	3.31	1.08	3.79	1.02	3.13	1.02
	Side effects	3.00	1.15	3.52	1.02	2.93	1.22
	Additional tasks	2.69	1.14	3.23	0.92	2.87	1.36
	Failure to meet requirements	3.50	1.10	3.38	0.85	3.24	0.83
	Lack of provider support	3.06	1.29	3.91	0.92	3.78	1.26
	Difficult to select	3.19	1.28	3.14	0.89	3.19	1.11

Table 5-5. Benefits and risks of COTS-based systems

Among the risks of building systems from COTS software, the UK SMEs indicated the difficulty of understanding the internal technical capability of COTS software. This is because COTS software products are considered as “black boxes” and complex (Vidger *et al*, 1996). On the other hand the UK software houses indicated the lack of COTS software provider support. Braun (1999) also points out that COTS software systems pose a risk, where the vendor goes out of business or fails to support the product. However, the Zambian organisations pointed out that the major risk with CBS is the periodic releases of COTS software. Carney and Wallnau (1998) found that rapid product release cycles imply schedule-driven use of COTS software components, normally at the expense of product stability and system stability. Remaining with the older versions of COTS products might also lead to interoperability problems with upgrades to other systems and may lead to a lack of support from the COTS software provider (Fox *et al*, 1997; Boehm and Abts, 1999). The relatively high standard deviation in the rating of risks associated with CBS among the UK SMEs and the Zambian organisations indicate that perceptions varied considerably from respondent to respondent.

The next section describes the findings related to the requirements engineering phase of CBS approach.

5.5.2 Requirements engineering phase

The respondents were asked to rate their strength of agreement to some factors related to building systems from COTS software. The questionnaire consisted of scaled-response from 1 to 5 such that 1 = *Never* and 5 = *Always*. This section presents the results regarding the techniques for acquiring and specifying requirements, identification of COTS software components, evaluation and selection of COTS software, COTS component evaluation criteria and organisational factors.

a. Techniques for acquiring and specifying requirements

Requirements engineering covers all of the activities involved in discovering, documenting and maintaining a set of requirements for a computer-based system (Sommerville and Sawyer, 1997). Requirements are defined during the early stages of system development as a specification of what should be implemented. Table 5-6 shows that in all the three samples the most frequently used techniques for requirements elicitation and specification are observation, prototyping and demonstrations. In prototyping the stakeholder is asked to comment on a prototype physical-working model of the desired system (Maiden and Rugg, 1996). Observation is a technique whereby the developer spends time in the working environment to observe the participants and elicit their requirements. However, the large standard deviation indicates considerable differences in the rating among the respondents in the three samples.

Although the importance of generating scenarios and use-cases and matching them to software component has been emphasised in literature (Maiden and Ncube, 1998), this was not highly rated by the study. It is also interesting to note that the techniques advocated by social-technical approaches such as rich pictures and Soft Systems Method (SSM) conceptual models scored very lowly in this survey. Bell and Davis (1999) argue that the use of SSM rich pictures and modelling of relevant systems (combined with stakeholder analysis) can help to surface ethical issues. Similarly, Avgerou and Cornford (1998) points out that the main value of SSM is in making analysts and other participants aware of unstructured organisational problems. The results from this survey suggest that organisations may not be adequately considering the social and organisational issues during requirements elicitation and specification.

Factors investigated	**SME (UK)**		**UK soft houses**		**Zambia**	
	Mean	**S.D.**	**Mean**	**S.D.**	**Mean**	**S.D.**
Data dictionary	2.08	1.38	2.90	1.12	3.75	1.34
Data flow diagrams	2.54	1.45	3.00	1.08	3.65	1.37
Decision trees and tables	1.77	0.93	2.22	1.08	2.38	1.26
Entity life cycles	1.85	1.21	2.51	1.21	3.33	1.45
ER modelling	1.77	1.09	2.39	1.14	3.38	1.59
Matrices	2.23	1.36	2.38	1.33	1.94	1.06
Meeting and interview	3.54	1.45	3.33	1.26	3.33	1.33
Normalisation	2.00	1.29	2.46	1.23	3.81	1.28
Object and class diagrams	2.08	1.26	2.80	1.12	2.13	1.19
Observation	3.69	1.25	3.56	1.05	4.00	0.94
Prototyping and demonstrations	3.69	1.18	3.79	0.89	3.88	1.17
SSM rich pictures	2.15	1.34	1.93	0.86	2.00	0.76
Scenarios	2.62	1.39	2.71	1.13	2.63	0.89
SSM conceptual models	1.77	1.17	1.78	0.88	2.53	1.25
Structure diagrams	2.54	1.56	2.83	1.27	2.80	1.26

Table 5-6. Requirements acquisition and specification techniques

Table 5-6 shows that, on average, the techniques used for eliciting user requirements (e.g. observation and prototyping) were rated more highly than the techniques used for modelling requirements (e.g. ER modelling, object and class diagrams). This suggests that, when building systems from COTS software, organisations elicit high level requirements but do not necessarily model these requirements before selecting COTS software components. This confirms the recommendations made by Dean and Vidger (1997), that in building systems from COTS software, organisations should acquire high-level requirements prior to an iterative and concurrent product evaluation and selection.

b. COTS component evaluation criteria

With regard to evaluation criteria, table 5-7 shows that, on average, software qualities were considered to be the most important criteria for evaluating COTS software by respondents from the UK SMEs and Zambian organisations. The importance of quality attributes in evaluating COTS software has already been identified by researchers in literature (Boloix and Robillard, 1995; Kontio, 1996; Maiden and Ncube, 1998). However, the results indicate that the UK software houses are more concerned with compliance with customer requirements. The relatively small standard deviation associated with rating software qualities (UK SMEs and Zambia) and compliance with requirements (UK software houses) suggest that there was general agreement among the respondents for each of the three samples.

Factors investigated	SME (UK)		UK soft houses		Zambia	
	Mean	S.D.	Mean	S.D.	Mean	S.D.
Ability to be tailored	2.69	1.38	3.28	0.80	3.13	1.26
Availability of documentation	3.92	0.67	4.12	0.70	4.41	0.87
Compliance with requirements	3.77	1.24	4.47	0.63	4.41	0.62
Conformance to appropriate standards	2.54	1.45	3.53	1.03	3.94	1.12
Ease of integration	2.31	1.11	3.43	1.06	3.00	1.46
Ease of migration	3.15	0.90	3.29	0.86	4.29	0.59
Existing relationship with supplier	2.92	1.44	3.23	1.02	3.06	1.24
Level of COTS supplier support available	4.00	0.71	4.07	0.67	4.18	0.95
Maturity of COTS products	3.42	1.16	4.07	0.84	4.29	0.77
Maturity of technology	3.46	1.05	3.91	0.92	4.06	0.85
Political and economic factors	1.77	1.24	2.14	0.81	2.31	1.08
Price of the COTS software product	3.69	0.95	3.70	0.77	4.38	0.72
Software qualities	4.17	0.58	4.33	0.75	4.59	0.62
Stability of COTS supplier	3.92	1.04	4.19	0.66	4.06	1.00
Viability of technology	3.69	1.18	4.12	0.70	3.94	1.03

Table 5-7: Evaluation criteria

c. Identification of COTS software components

In CBS it is important to identify COTS software components that meet the high level customer requirements, which can then be considered for a more rigorous evaluation. Table 5-8 shows that both SMEs and software houses from the UK consider customer prior knowledge and experience as an important technique for identifying COTS software components. The respondents from Zambia indicated that an inventory of COTS software components within the organisation is the most important activity in the identification of software components. The second most important factor for Zambian organisations was customer prior knowledge and experience. The results suggest that there are problems with techniques for identifying COTS software products because the most highly rated techniques such as inventory and prior knowledge do not take into consideration new products in the marketplace.

Factors investigated	SME (UK)		UK soft houses		Zambia organ.	
	Mean	S.D.	Mean	S.D.	Mean	S.D.
Adverts and promotions	2.92	1.26	3.02	0.94	3.38	1.02
Fairs and shows	2.77	1.01	2.80	0.90	2.73	1.10
Internet (Web) search	2.75	1.22	3.36	0.91	3.00	1.37
Inventory of existing COTS	2.15	0.99	2.95	1.03	3.88	0.99
Market research	3.00	1.29	3.29	1.15	3.47	1.51
Prior knowledge and experience	3.54	0.88	3.58	0.93	3.50	1.29
Request For Proposals (RFPs)	2.23	1.36	2.71	1.07	2.71	1.26

Table 5-8. Techniques for identifying COTS components

d. Evaluation and selection of COTS software

Table 5-9 indicates that on average the most frequently used technique by UK software houses and Zambian organisations for evaluating and selecting COTS software is studying the documentation. Attending vendor demonstration was the most highly rated technique by the UK SMEs. This is consistent with the findings of Tran *et al* (1997) that COTS software evaluation involves extensive effort and time reviewing literature, attending conferences (demonstrations), travelling and communicating with product vendors.

Carney and Wallnau (1998) argue that the COTS software evaluation and selection is a form of decision making. Kontio (1996), Maiden and Ncube (1998) points out that it is a multi-attribute decision making process. It is therefore surprising to note that multi-criteria decision-making techniques such as outranking and AHP, which have been advocated in literature (see section 3.3.4), were not highly rated by the respondents. Multi-criteria decision making techniques help the decision maker to structure the decision problem. The low rating by the respondents suggests that COTS software evaluation and selection have been conducted in an "ad-hoc" manner by some of these organisations. This is consistent with the findings of Kontio (1996) that most organisations are under pressure to perform and therefore do not use a well defined repeatable process. The evaluators do not have the time or experience to plan the selection process in detail, and therefore, they do not use the most appropriate methods in the selection process.

Factors investigated	**SME (UK)**		**UK soft houses**		**Zambia**	
	Mean	**S.D.**	**Mean**	**S.D.**	**Mean**	**S.D.**
Analytic Hierarchy Process (AHP)	1.58	0.79	1.60	0.87	1.93	1.00
Attend demonstration	3.69	1.38	3.79	0.80	3.71	1.31
Cards sorting and laddering	1.62	1.19	1.53	0.72	1.60	0.63
Customer experience	3.15	0.69	3.33	1.05	3.53	1.28
Extensive experimentation	3.08	1.26	3.62	0.88	3.00	1.25
Feature analysis	2.23	1.36	2.33	1.12	2.81	1.60
Multi-criteria decision making	1.77	1.09	1.93	1.17	2.07	1.16
Outranking	1.62	0.77	2.02	1.13	2.07	1.21
Study documentation	3.54	1.05	3.93	0.87	4.18	1.24
User community experience	2.92	0.86	3.40	0.85	3.56	1.21

Table 5-9. Techniques for evaluation and selection of COTS software

e. Organisational factors

Table 5-10 indicates that, on average for the UK SMEs, the most important organisational factor considered when building systems from COTS software is customer motivation. Davis and Olson (1985) point out that considering individual motivation when developing systems is difficult because determining individual motivational needs are very subjective and often inaccurate.

However, for the UK software houses the most important organisational factor for CBS development is customer education and training. Lack of proper customer training and documentation are known to lead to systems failure (Laudon and Laudon, 1998). Davis and Olson (1985) points out that appropriate user training is an important factor in overcoming user resistance to new systems. Lynex and Layzell (1997) suggest that identification and provision of appropriate user training is an important activity that can help with the problems of low user education skills and knowledge.

Themes	Factors investigated	SME (UK)		UK soft houses		Zambia	
		Mean	S.D.	Mean	S.D.	Mean	S.D.
Organisational factors	Customer motivation	3.08	1.44	3.16	1.11	3.47	1.28
	Customer education & training	2.85	1.46	3.37	1.18	3.89	1.32
	Group communication	2.31	1.03	2.98	1.14	3.35	1.32
	Organisation structure &politics	2.23	0.93	2.79	1.21	3.76	1.15
	Changing business strategy	2.54	1.27	2.95	1.15	3.88	1.11
	Organisation resource &support	2.85	1.52	3.19	1.11	4.00	1.12
	Organisational culture	2.54	1.45	2.83	1.21	3.47	1.07
	External factors	2.38	1.19	2.68	1.19	3.60	1.30
Customer participation in these stages	Requirements acquisition	2.77	1.48	3.63	1.43	3.80	1.32
	Requirements specification	3.08	1.66	3.74	1.45	3.88	1.32
	Systems & architecture design	1.85	0.99	2.81	1.35	2.82	1.42
	COTS identification	2.00	1.29	2.56	1.16	3.00	1.12
	COTS evaluation	2.23	1.36	2.84	1.38	3.29	1.49
	COTS selection	2.08	1.44	2.79	1.36	3.06	1.56

Table 5-10. Organisational factors

The Zambian organisations considered organisational resources and support as the most important factor for successful development of systems from COTS software. This is consistent with literature that organisations in DCs suffer from a lack of resources and this has a negative impact on successful development and implementation of software systems (Janczewski, 1992). Lynex and Layzell (1997) point out that securing management support is critical to the successful implementation of any method incorporating reusable software components. Therefore, educating management by using incremental approach and demonstrating to them successful case studies would help to elicit management support. Furthermore, the technical jargon related to CBS benefits should be reduced into a simple to understand business case.

Taylor and Felten (1993) argue that participation of members is essential, not only for acceptance and commitment to the new design but more importantly to develop and build the new system, taking into account the human and social issues. The respondents were asked to indicate in which stage of the requirements engineering phase they have a stronger preference for customer participation. The respondents from the three samples indicated their preference for customer participation in the earlier requirements acquisition and specification compared to later part of the system development, such as the design stage. However, the relatively large standard deviation indicates that there were considerable differences in their preferences.

5.5.3 Approaches to building systems from COTS

The primary approaches to building systems from COTS software identified in literature are the buy and use; buy and adapt; and the component integration (Brown and Wallnau, 1996b; Vigder *et al*, 1996). The respondents were asked to select all applicable factors (answers) from a list of factors in response to questions regarding their experiences with CBS approaches. For example, an organisation may be using all the three approaches or only using "buy and adapt" approach.

Table 5-11 indicates that, on average, the most frequently used CBS approach by the UK SMEs is the buy and use model. In the buy and use model a single complete working COTS software system that satisfies most of the user requirements is purchased and used without adapting or extending it (see section 3.2.1). According to the ratings, the most frequently used CBS approach by Zambian organisations is purchase-and-adapt model. The buy-and-adapt model is characterised by acquiring a single complete working system that satisfies most of the requirements of the acquisition agency and adapting or extending it for local needs (see section 3.2.4c).

Themes	**Factors investigated**	**UK SME (%)**	**UK Software Houses (%)**	**Zambia (%)**
CBS approaches	Purchase and use	31.3	50.9	40.0
	Purchase and adapt	21.9	49.1	75.0
	Integrate	25.0	57.9	55.0
Purchase and use	Office automation	31.3	38.6	55.0
	Database systems	21.9	43.9	45.0
	Accounting and finance	15.6	28.1	50.0
	Email and messaging systems	21.9	35.1	50.0
	GUI builders	6.3	22.8	15.0
	Geographic Information Systems	0.0	12.3	5.0
	Operating systems	28.1	43.9	45.0
	Real time & embedded systems	9.4	17.5	5.0
	Safety critical systems	0.0	12.3	0.0
	Business applications	9.4	24.6	35.0
Problems with purchase and use approach	Not satisfying requirements	18.8	24.6	40.0
	New releases of COTS software	12.5	38.6	35.0
	Lack of COTS provider support	25.0	35.1	30.0

Table 5-11. CBS approaches

On the other hand, the results show that the most frequently used approach by the UK software houses is the "component integration" approach. The component integration model involves purchasing a number of off-the-shelf components, each satisfying some part of the requirements of the system, and integrating these components into the required system (see section 3.2.4d). This approach is considered to be the "ultimate" stage in building systems from COTS software. Therefore the inference that can be drawn is that CBS is more mature in the UK software houses than in the UK SMEs and Zambian organisations.

a. Purchase and use approach

Table 5-11 shows that, on average, the main application of purchase and use approach by Zambian organisations and the UK SMEs are office automation. However, the respondents from the UK software houses indicated that the main application for this approach is databases. This suggests that there is greater availability and maturity of COTS products in this application domain (Vigder *et al*, 1996; Dean and Vigder, 1997). It is interesting to note that the UK SMEs and Zambian respondents did not indicate purchasing COTS software and applying them to safety critical systems. This agrees with Brown and Wallnau's (1996b) argument that using COTS software components to build safety-critical systems where reliability, availability, predictability and security are required is too risky and difficult.

Regarding the problems experienced with purchase and use approach, Table 5-11 shows that the most significant problem with the Zambian organisation is that COTS software does not meet requirements. This is in agreement with Braun (1999) who argues that COTS software may not match performance parameters because they are not tailor-made for a specific organisation. The UK SMEs respondents indicated that the most significant problem with this approach is the new releases of COTS software. However, the respondents from the UK software houses rated the lack of COTS provider support as the most significant problem with this approach.

b. Purchase and adapt approach

The majority of the respondents from the UK indicated that the main application of the purchase and adapt approach is database systems, while the Zambian organisations indicated accounting and finance (see table 5-12). This indicates the application domains in which COTS products are mature. Again, as in the purchase-and-buy model, Zambian respondents indicated that they do not purchase and adapt COTS products in the safety-critical domain.

Themes	Factors investigated	SME (UK) (%)	Software Houses (UK) (%)	Zambian Organ. (%)
Main applications	Office automation	6.3	14.0	15
	Database systems	25.0	33.3	45
	Accounting and finance	15.6	21.1	50
	Email and messaging systems	3.1	5.3	15
	GUI builders	0	10.5	20
	Geographic Information Systems	0	8.8	5
	Operating systems	6.3	19.3	0
	Real time and embedded systems	9.4	21.1	0
	Safety critical systems	3.1	15.8	0
	Business applications	6.3	21.1	40
Programming languages or development tools	Java/Javascript	6.3	12.3	10
	VisualBasic	15.6	33.3	40
	Applescript	0	5.3	5
	Perl	0	8.8	0
	Ada	0	12.3	0
	C/C++	15.6	42.1	25
	Delphi	3.1	10.5	0
Techniques for adapting	API	6.3	38.6	55
	Modify source code	3.1	26.3	30
	Plug-ins	9.4	40.4	10
	Scripting	15.6	31.6	25
	Inheritance	0	15.8	10
Problems experienced	Limited choice of supply of COTS	3.1	15.8	10
	Not satisfying requirements	6.3	21.1	20
	New releases of COTS	9.4	29.8	15
	High prices	6.3	12.3	5
	Lack of COTS provider support	6.3	26.3	30
	Difficult to modify COTS	15.6	29.8	45

Table 5-12. Purchase and adapt CBS approach

The majority of the respondents from the UK SMEs indicated that VisualBasic and C/C++ are the main programming languages used to adapt the products (see table 5-12). On the other hand the UK software houses highly rated C/C++ as the main programming language while the Zambian organisation indicated Visual Basic. Visual Basic programming language can be easily used to tailor most of the COTS components sold by Microsoft, such as Office2000. It is also interesting to note that Zambian organisations lowly rated other programming languages and development tools (e.g. Java/Javascript, Applescript, Perl, Ada). This explains the limited application of buy-and-adapt model and the difficulties to modify COTS software products experienced by the Zambian organisations (see table 5-12). Therefore, Zambian organisations would greatly benefit from CBS when they strengthen capacities in these programming languages and development tools.

The respondents did not agree regarding the techniques for tailoring or adapting COTS software components (see table 5-12). The majority of respondents from the UK SMEs highly scored scripting while the UK software houses indicated writing plug-ins. The majority of respondents from Zambian organisations indicated the use of the component APIs to adapt COTS components. The advantage of using the APIs is that most of the COTS software products available in the market have some kind of an API that can be easily accessed by a developer.

Regarding the problems experienced with this approach, table 5-12 indicates that all the respondents agreed that the most significant problem is the difficulty to modify or adapt COTS components. The problems related to the difficulty to modify COTS software include of lack of access to COTS software internals and non-conformity to standards (Vigder *et al*, 1996; Braun, 1999). The UK software houses also brought out the problem of new releases of COTS software. This is consistent with Boehm and Abts (1999) who points out that the vendors do not normally support old releases of their COTS software products and upgrades may not be compatible with old releases.

c. Component Integration approach

The majority of the respondents indicated that the components normally integrated are off-the-shelf applications and system services like databases or operating systems. There are several methods and mechanisms for integrating COTS components. Table 5-13 shows that the most highly rated method for integrating COTS software was procedure calls (SMEs and software house) and data sharing (SMEs and Zambia). Examples of procedural calls include components that are packaged as a procedural library, applications with an API, and databases with an SQL interface (Vigder *et al*, 1996; Dean and Vigder, 1997). In data sharing, integration can be accomplished by having multiple components share a common data repository reading and writing the same data objects (Rader, 1997).

Although the literature regards CORBA as an important standard for COTS component integration (Vigder *et al*, 1996; Dean and Vigder, 1997) these were not highly rated in this study. In contrast, table 5-13 shows that the majority of the respondents indicated Object Linking and Embedding (OLE) and DDE as the most significant technologies for integration. Both DDE and OLE technologies are part of COM technology developed by Microsoft Corporation (see section 3.2.4).

Themes	Factors investigated	SME (UK) (%)	Soft Houses (UK) (%)	Zambia (%)
Components normally integrated	Procedural libraries	12.5	22.8	20
	Legacy applications	0	22.8	5
	Off-the-shelf applications	12.5	28.1	35
	Tools e.g. GUI builder	9.4	22.8	25
	System services like database or OS	18.8	35.1	40
	Frameworks	0	8.8	0
	OLE objects	6.3	24.6	15
Methods for integrating components	Procedural calls through API	18.8	42.1	40
	Desktop supported facilities	9.4	26.3	40
	Message bus	0	14	5
	Data sharing	18.8	33.3	50
	Object request broker	0	7	0
Standards and technologies	COM	6.3	22.8	15
	DCOM	3.1	10.5	5
	OLE	9.4	36.8	40
	DDE	15.6	29.8	35
	ActiveX	6.3	17.5	5
	CORBA	0	8.8	0
	OpenDOC	0	1.8	5
	OSA	0	3.5	5
	RMI	0	3.5	0
Main applications	Office automation	9.4	14	15
	Database systems	9.4	24.6	25
	Accounting and finance	6.3	12.3	40
	Email and messaging systems	0	5.3	15
	GUI builders	3.1	8.8	10
	Geographic Information Systems	0	5.3	5
	Operating systems	6.3	12.3	5
	Real time and embedded systems	3.1	17.5	5
	Safety critical systems	3.1	8.8	0
	Business applications	12.5	21.1	30
Problems experienced	Lack of support from COTS provider	9.4	21.1	20
	New releases of COTS	9.4	26.3	5
	Lack of information about COTS	6.3	19.3	35
	Difficult to integrate components	12.5	28.1	30
	Conflicting standards	0	26.3	40

Table 5-13. Components Integration model

The respondents in the three samples did not agree regarding the main COTS software applications used in the integration model; the UK SMEs indicated business applications while the UK software houses indicated database systems. On the other hand respondents from Zambian organisations rated accounting and finance as the main applications for the integration model. Most of the commercial accounting and financial applications are provided in modules with an API.

Table 5-13 shows that the majority of respondents from the UK (SMEs and software houses) agreed that the main problem experienced with this approach is the difficulty with integrating COTS components. This is consistent with Boehm and Abts (1999) who argues that integrating software components is difficult because most COTS products are not designed to interoperate with each other. In contrast, Zambian organisations indicated that most significant problem with integrating COTS software is conflicting standards. Haines *et al* (1997) points out that the degree to which a software component meets certain standards greatly influences the interoperability and portability of a system. The findings therefore indicate that interoperability is an important factor when building systems from COTS software products.

The next section discusses the differences between the UK and Zambia regarding building systems from COTS software.

5.5.4 Differences between the UK and Zambian sample

This section presents results of the assessement of the differences between the UK and Zambia regarding building systems from COTS software. The level of significance was set at 5% ($p = 0.05$). Initially the Kruskal-Wallis test was used to assess the differences between the three samples (UK SMEs, UK Software houses and Zambian organisations). However, as indicated in section 4.4.5, the Scheffe test (see table 5-14) was used to assess the differences between Zambian organisations and the UK.

Table 5-14 shows that there is a difference between Zambian organisations and the UK software houses regarding lack of time as a constraint of developing software systems. It appears that "lack of time" is considered significantly more important by the UK software houses compared with the Zambian organisations. Among the main benefits of CBS, table 5-14 shows that there are differences between Zambian organisations and the UK regarding the perception of benefits of CBS, for example to improve reuse among projects.

Themes	Dependent Variable	Comparison to Zambian sample	Mean Difference	Std. Error	Sig.
Constraints or obstacles	Lack of time	UK SME	-0.548	0.340	0.335
		UK Software House	-1.053	0.297	0.002*
CBS Benefits	Improves reuse	UK SME	0.824	0.315	0.032*
		UK Software House	0.663	0.263	0.041*
	Competitive marketplace	UK SME	0.647	0.325	0.151
		UK Software House	0.756	0.272	0.021*
Requirements engineering phase	ER modelling	UK SME	1.518	0.457	0.004*
		UK Software House	0.946	0.367	0.036*
	Normalisation	UK SME	1.670	0.463	0.002*
		UK Software House	1.313	0.372	0.002*
	Data dictionary	UK SME	1.536	0.448	0.003*
		UK Software House	0.843	0.359	0.065
	Entity life cycles	UK SME	1.548	0.469	0.005*
		UK Software House	0.786	0.379	0.126
	Data flow diagrams	UK SME	1.147	0.435	0.031*
		UK Software House	0.647	0.346	0.196*
Identifying COTS software	Inventory of existing COTS	UK SME	1.811	0.365	0.000*
		UK Software House	0.929	0.290	0.006*
Evaluation criteria	Price of the COTS software product	UK SME	0.708	0.282	0.042*
		UK Software House	0.670	0.229	0.014*
	Ease of migration	UK SME	1.094	0.285	0.001*
		UK Software House	1.015	0.230	0.000*
	Conformance to appropriate standards	UK SME	1.471	0.405	0.002*
		UK Software House	0.415	0.329	0.636
Organisational factors	Organisation structure and politics	UK SME	1.408	0.422	0.004*
		UK Software House	1.015	0.334	0.010*
	Changing business strategy	UK SME	1.239	0.425	0.014*
		UK Software House	0.975	0.338	0.015*
	Organisational resources & support	UK SME	1.143	0.434	0.037*
		UK Software House	0.860	0.345	0.051
	External factors	UK SME	1.243	0.450	0.027*
		UK Software House	0.966	0.366	0.036*

* significant at 0.05 or greater

Table 5-14. Scheffe test showing the differences between Zambia and the UK

It is interesting to note from table 5-14, in the requirements engineering phase, the differences between the UK and Zambian samples are those related to user requirements modelling. The survey indicates that the differences between Zambian organisations and SME are attributed to ER modelling, normalisation, data dictionary, entity life cycles and data flow diagrams. The difference between the Zambian organisations and the UK software houses are attributed to ER modelling and normalisation. The results also show that there is no significant difference in the perception of the three samples regarding most of the requirements elicitation techniques (meeting and interview with users, observation, attending demonstrations and prototyping).

One of the insights from table 5-14 is that there are significant differences between the UK and Zambian organisations regarding using the inventory of existing COTS software for identifying COTS software. Furthermore, table 5-8 indicates that Zambian organisations consider this technique to be the most important activity in COTS software identification. This suggests that Zambian organisations lack the necessary resources to use other techniques such Internet search and market surveys to identify all the potential COTS software products in a particular application domain.

Regarding the COTS software evaluation criteria, the survey shows that there were differences in the ratings between the Zambian organisations and the UK SMEs regarding the importance of cost of the COTS software, ease of migration, and conformance to appropriate standards. Furthermore, there was a difference between the Zambian organisations and the UK software houses regarding the importance of including the cost of the COTS software and ease of migration in the evaluation criteria. The Zambian organisation perceived that the cost of the COTS product and the ease of migration to a similar product from a different vendor were very important when purchasing COTS software.

Table 5-14 also shows that there were differences between the average rating of the three samples regarding the important organisational factors in CBS. The survey findings indicate that the differences between Zambian organisations and the UK samples were organisational structure and politics; changing business strategy; and external factors. This suggests that there are cultural differences between the UK and Zambia. However, regarding the importance of organisational resources and support, the differences between Zambian organisations and the UK software houses was not significant.

5.6 Discussion of findings

This survey was aimed at eliciting and synthesising current practices for building systems from COTS software from the UK and Zambia. The results of the survey suggest that on average there are *similarities* between the UK and Zambia regarding CBS practices. These include that:

- The main *constraint or obstacle to developing software systems* in the UK SMEs and Zambia *is the lack of adequately trained human resources*. This supports the findings already identified in literature that there is shortage of IT personnel in DCs (Woherem, 1992a; Corr, 1995) and in UK SMEs (Spectrum, 1998). This suggests that to fully benefit from CBS approaches, organisations must be willing to invest their resources in human resource development and training.
- The most significant *benefit of CBS* in the UK and Zambia *is in reducing software development cost*. This indicates that organisations in the UK and Zambia can benefit from building systems from COTS components by reducing the software development costs. Building systems from COTS components is cheaper because the essential requirements need not be specified in detail (as with bespoke systems) and the cost of COTS component is shared among a number of users.
- Regarding the *requirements engineering phase*, the most frequently used techniques in both the UK and Zambia *are observation, prototyping and demonstrations*. Furthermore, the techniques used for eliciting user requirements were rated higher than the techniques used for modelling requirements. The findings corroborate the recommendations from the literature that, when building systems from COTS software, organisations should not spend too much effort in defining to the lowest detail the desired characteristics of the required system (Oberndorf, 1997). In other words, organisations should only acquire high level requirements and then purchase the products from the marketplace that match these high level requirements. In this way organisations will be able to benefit from CBS approaches by lowering the development costs.
- During COTS software evaluation and selection, the UK software houses and Zambia rated *studying documentation as the most significant technique for evaluating COTS software*. However, the problem with relying on studying documentation is that the selected product might not be compatible with other existing software. Therefore, the findings also suggest that there are problems with

the COTS software selection process. Furthermore, the low rating by the three samples of systematic approaches to COTS software selection such as market survey and multi-criteria decision-making techniques further augment the suggestion that COTS software evaluation and selection is problematic both in the UK and Zambia.

- Regarding the COTS software evaluation criteria, the UK SMEs and Zambia rated *software quality attributes as the most frequently used criteria*. Software quality attributes include portability, usability, efficiency, scalability and dependability. It was interesting to note that in the UK and Zambia, political and economic factors were the lowest rated. This is not surprising because, although organisations are aware of political influences, it is difficult to include them in the COTS software evaluation criteria. Researchers have recommended customer participation as an effective strategy for improving software design outcomes and as a means of incorporating human and organisational aspects (Gould *et al*, 1991; Bravo, 1993; Axtell *et al*, 1997). This suggests organisations should include customers in the process of COTS software evaluation and selection to address these organisational issues.
- Contrary to the majority of articles about the importance of CORBA technology in COTS integration, this study shows that the most significant *technology used by practitioners* in the UK and Zambia *is Microsoft's OLE and DDE*. However, OLE and DDE have limited to Windows operating systems and are not portable to other platforms (Sessions, 1998). Therefore, to realise the full potential of CBS, organisations should invest their resources in other component integration technologies such as CORBA and Enterprise JavaBeans (see section 3.2.4).

Comparing the Zambian organisations to the UK organisations the following *differences* were observed:

- There are statistically significant differences between Zambian organisations and the UK *regarding the problem of lack of time in building software systems*. Organisations in the UK consider time as an important resource in software development. This suggests that there are cultural and contextual differences. Bjorn-Andersen (1990) and Janczewski, (1992) argue that techniques and software systems developed with different socio-cultural context must be adapted when applied to DCs. Therefore, systems and frameworks developed in the UK must take into consideration these differences when applied in DCs such as Zambia.

- Among the *benefits of building systems from COTS software*, significant differences were observed between Zambian organisations and the UK regarding *whether CBS improves reuse across projects and whether it offers competitive marketplace*. The higher rating by Zambian organisations, compared to the UK, suggest that DCs could draw more benefits from CBS.
- There are significant differences between Zambian organisations and the UK regarding *techniques for modelling customer requirements*. The higher rating by the Zambian organisations compared to the UK indicates that Zambian organisations are spending too much effort in defining to the lowest detail the desired characteristics of the required system. This has been identified in literature (Maiden and Ncube, 1998), whereby using COTS software to build systems presents new problems for the requirements engineer. This suggests that Zambian organisations can learn from the UK on how to model user requirements for CBS.
- Based on the findings, *an inventory of existing COTS software was the only technique for identifying COTS software* on which the respondents from Zambian organisations and the UK significantly differed. Further, *there were differences of opinion between what to include in the evaluation criteria*. The significant differences observed were regarding the price, ease of migration and conformance to appropriate standards. The problem associated with the evaluation criteria has been identified in literature. For example, Kontio (1996) argued that evaluators sometimes include immaterial and inappropriate attributes in the criteria leading to incompatibilities. The differences between the Zambian organisations and the UK regarding the evaluation criteria and the techniques for identifying COTS software provides some problem areas that require investigation.
- Regarding organisational factors, the results indicate that there are differences between Zambian organisations and the UK. The differences observed were about the rating of *organisation structure and politics, changing business strategy, organisational resources and support, and external factors*. This also suggests that there are social-cultural differences between Zambia and the UK. Therefore, the significant differences in the organisational factors provide important aspects to consider when adapting the framework developed from the UK for application in

Zambia.

- The most frequently used *CBS approach for UK software house is an integration approach, while for Zambian organisations it is a purchase and adapt approach*. Further, the results indicate that few organisations in Zambia were familiar with CORBA and Enterprise JavaBeans technology compared to the UK. This suggests that the maturity of CBS in the UK software houses is higher compared with Zambian organisations. Therefore, Zambian organisations can learn and adapt the best practices from the UK regarding CBS.

The value of the results of this study is that an understanding of the development process, benefits and risks associated CBS has been elicited. Furthermore, a better understanding of the technological problems facing Zambia was obtained. This resulted in appreciating how CBS approaches could provide better ways of helping these organisations develop and implement more effective information systems. The results also brought out significant differences between the UK and Zambia regarding CBS practices. Therefore, the results from this survey assisted in later adapting the framework developed from the UK data in order to apply it to a developing country, Zambia (see section 7.3). The study also helped in identifying and making contact with organisations that would participate in further research.

The limitation of this survey was that it focussed on one developing country, Zambia. Therefore generalisations about the findings cannot easily be made to other DCs with different characteristics, since some of these findings could be tightly related to factors that are peculiar to Zambia or Zambian organisations. However, the similarities and differences between the UK and Zambia provided adequate insight regarding the problems of COTS software evaluation to get the research started.

5.7 Summary

This chapter presented the findings of the first empirical study aimed at eliciting and synthesising current CBS practices from the UK and Zambia. A survey approach was adopted for study through the administration of self-completion questionnaires. The sample comprised of thirty-six respondents from UK SMEs, fifty-nine respondents from UK software houses and twenty respondents from Zambian organisations. The respondents were mainly managers, systems analysts, programmers and academics experienced in CBS approaches.

A number of lessons were learnt by comparing the Zambian organisations with the UK organisation. For example, both the UK software houses and Zambia rated studying documentation as an important technique for COTS software evaluation and selection, suggesting problems with COTS software evaluation and selection practices. This is because evaluators might select a product that is not compatible with other software by using documentation alone without experimenting with the product. The survey also brought out differences between the UK and Zambia. For example, there were differences of opinion regarding what to include in the COTS software evaluation criteria, suggesting that some evaluators might be including immaterial and inappropriate attributes in the criteria.

As a result of this survey a better understanding of how CBS can provide support for organisations in DCs, such as Zambia, to develop and implement more effective information systems was achieved. This survey brought out a number of problems associated with building systems from COTS software. In particular, the problem of COTS software evaluation and selection to support the CBS process was highlighted, resulting in a more focussed direction for the research project. Therefore, the second study was aimed at identifying processes (including traditional and soft factors) that support COTS software component selection for CBS. The findings of the second study and its significance on this research are presented in the next chapter.

6 Identifying factors that support COTS software selection (study 2)

This chapter presents the findings of the second study aimed at identifying important processes and factors that support COTS software selection. The chapter also presents a review of the research method used for this study. A high level overview of the resulting framework (STACE) is provided in this chapter. Finally, the chapter discusses the significance of the findings on the overall research.

6.1 Introduction

The overall goal of the second study is to identify important processes (including traditional and soft factors) that support COTS software component selection for CBS from organisations in the UK (see section 1.4). A process is a collection of related tasks leading to a product, for example requirements definition process comprise a number of tasks (and activities) resulting in requirements documents. A factor is a circumstance or influence contributing to a result, for example cost is an important factor in COTS software selection. Therefore a number of factors can be associated with a process.

The first study aimed at eliciting CBS practices brought out the problem of COTS software evaluation and selection (see section 5.6). Therefore, this study focusses on identifying important processes and factors that support COTS software selection. These processes and factors are considered essential by experts in the field to minimise the risks and address problems of COTS software selection discussed in section 3.5.1. For example, the lack of a well defined COTS software selection process will lead in the use of inappropriate methods while neglect of non-technical factors such as vendor capability can lead into selecting a product from a vendor that runs out of business. Identifying these processes and factors will also assist in defining how an organisation is supposed to perform its activities related to COTS software selection, and how people work and interact.

The research questions investigated during this study are:

- What are the most important processes (techniques, tools and other factors) that support COTS software evaluation and selection?
- How can these important processes and factors be classified and how do they relate to each other?
- How can the social-technical approach be used to improve COTS software evaluation and selection? In particular, in what ways might customer participation and use of social-technical criteria contribute to COTS components evaluation success?
- What kinds of problems (and solutions) have organisations experienced in evaluating COTS components for CBS?

The outcome of this study led to the development of a framework for selecting COTS software that incorporates the often-neglected non-technical issues.

6.2 Research Method

A field study approach comprising a set of interviews was used for this study, allowing cross-organisation analysis and comparison, which is important for identifying patterns and developing theoretical categories. Section 4.5 provides a more detailed discussion of the research method and the rationale for adopting this approach. An interview protocol was developed to guide the researcher in carrying out the study and this helped to increase the reliability and sharpen the construct validity of the research. Theoretical sampling was used as the basis for selecting organisations for this study, to focus on organisations that confirmed, extended and sharpened the theoretical framework (see sections 4.5.1 and 4.5.5).

The general mode of analysis used in this study was explanation building. This involved five main steps: 1) Categorising, to identify concepts and develop coding categories around them; 2) Tabulating, to create tables for data display so that valid conclusions could be drawn; 3) Explanation, to provide explanations that validate the relationships between higher level coding categories; 4) Modelling, to build networks based on the relationships between higher level coding categories; 5) Review of findings, to review the draft field study reports by key informants so as to increase construct validity and facilitate selection of other organisations (see section 4.5.4).

Data analysis was supported by ATLAS/ti™ a qualitative software analysis tool, which also acted as field study database. The use of a software tool and field study database enhanced the reliability and validity of the findings. The rigorous nature of the data collection and analysis procedures provides confidence that the findings of this study are valid (see section 4.7 for more detailed discussion). The next section presents the field study organisations and their background information.

6.3 Field study organisations

A total of 16 in-depth interviews were conducted in 8 organisations within the UK. They were selected from organisations that participated in the first study (see section 5.4.1). They were included in this study because of their experience in CBS and a variety of systems development techniques, and it was perceived that they would provide literal and theoretical replication (Yin, 1994). Furthermore, deliberate effort was made to ensure that a wide variety of organisations from different sectors were included in the field study (see table 6-1).

Organisation #	Main business area	Size of organisation	Number of interviews
1	Software house	Small	1
2	Research and development	Medium	3
3	Local authority	Medium	2
4	Manufacturing/ Engineering	Large	3
5	Finance/ Banking	Large	2
6	Consulting services	Small	1
7	Retail/ Wholesale	Large	3
8	Telecommunications	Large	1

Table 6-1. Background information about participant organisations

Background information and the COTS software application domain of the 8 field study organisations is described in the remaining parts of this section.

a. Organisation 1 (Software house)

This organisation provides consulting services that enables people from across an organisation to work together, defining, communicating and improving the way their business works. This organisation develops software tools to help its customers develop their process, people and technology assets in a synchronised way, enabling them to deliver sustainable value. The organisation had an annual turnover of about £675,000 in 1997. The interview was conducted with a senior consultant responsible for project management, workflow systems and business modelling, and software development (integrating different modelling tools).

b. Organisation 2 (Research and development)

This organisation is one of the world's largest independent contract research, development and testing organisations. The organisation employs an expanding team of 500 research scientists, engineers and support personnel. In terms of software engineering the organisation has experience in software reliability; safety critical software; software testing; software development; Internet and Intranet technologies; systems integration. Three separate interviews were conducted with senior members of staff at this organisation. The applications of COTS software components that were investigated in this organisation were high integrity systems and multimedia systems.

c. Organisation 3 (Local authority)

This organisation provides services to 175,000 people including education, social services, leisure, highways, planning, cleaning and many more. The information Technology and telecommunications (IT) department provides support for computer systems throughout the authority. For example, the IT department advises the other departments on the process to follow when acquiring COTS software and also helps with evaluation by allocating IT staff to evaluating teams. The organisation gets funding through local taxes and government support. Two interviews were conducted with IT managers.

d. Organisation 4 (Manufacturing/ Engineering)

This organisation is a world leader in the power systems business, providing cost-effectively engineered products and services to commercial and military customers in propulsion, electrical power and materials handling markets around the world. It has a turnover of over £1 billion with over 42,000 employees and customers in 135 countries. Interviews were conducted with three senior members of staff in the department responsible for software development and maintenance.

e. Organisation 5 (Finance/ Banking)

This organisation offers a number of specialist business services including business banking, commercial banking, corporate and institutional banking in the UK and overseas. At the end of 1997 total Group assets for this organisation were £176 billion and there were over 76,000 employees. The IT department has over 1500 software developers dedicated to developing and maintaining software systems. Interviews were conducted with two persons from the Architecture Group, who are responsible for developing both business and software architectures for the whole organisation.

f. Organisation 6 (Consulting services)

This organisation, although small, is one of the market leaders in management and financial services consultancy. The management services specialise in software solutions for personnel, point of sales systems for sales forces, training, design and publishing of interactive media used in the sales process. The organisation has offices in Australia, Canada, Hong Kong, USA, Ireland, Malaysia, Spain, and South Africa.

The organisation has developed a number of commercial software including Trackrecord™, which is a personnel tracking system for financial services. Trackrecord™ is used by an increasing number of financial institutions, including some of the UK's largest banks, building societies and friendly societies. The interview was conducted with the managing consultant who was the key person in the development of this system.

g. Organisation 7 (Retailer/ Wholesaler)

This organisation is a leading retailer with over 370 stores in the United Kingdom, Europe, Hong Kong and Canada. It employs 68,208 people around the world and achieved a group turnover of over £8 billion. Most of the software development is carried out through outsourcing to software houses but the organisation still maintains a pool of its own analysts and developers.

Interviews were conducted with three senior members of staff from Stores System Project (SSP) and Contracts Management System (CMS). The SSP has adopted a functionality approach, where they identify the functionality that goes into the stores system and then componentise that. The CMS project involves management of contract with suppliers aimed at enhancing working relationships with them. However this excludes the suppliers of food stores. This is aimed at replacing paper-based systems for product development and contracting.

h. Organisation 8 (Telecommunications)

This organisation is one of the leading providers of telecommunication services in the UK. They have also a world-wide presence through a series of subsidiaries, equity ventures and distributorships in Europe, Asia Pacific and the Americas delivering a comprehensive multi-local strategy covering the datawave, the IP world, mobile, multimedia and fixed to mobile convergence. The organisation has over 1500 employees and a turnover of over £1 billion. The interview was conducted with one individual involved in the component-based development (CBD) project within the organisation.

The objective of the CBD project was to investigate the impact of component-based development when adopted in the organisation, in terms of whether they would have to change their processes. In particular, the difference it would make to the software development life cycle if software components where bought from outside and how they would be integrated with existing systems. This was perceived as important because the organisation currently has many different software systems and therefore using COTS software components would pose a number of integration challenges. The project comprised of seven experts from within the organisation.

6.4 Field study results

This section presents the cross-organisation analysis of the identified processes (including traditional and soft factors) that support COTS software component selection.

6.4.1 Requirements definition

The respondents brought a number of techniques and factors that they considered important during the process of defining customer requirements (see table 6-2). For example, organisations 3 and 4 brought out the importance of defining the problem to be addressed by the COTS software product before embarking on the evaluation process. This was supported by respondents from organisations 6 and 8 who argued that the high level requirements must be elicited from the stakeholders and domain knowledge. Respondents from organisations 3, 4, 5 and 8 pointed out the importance of performing business appraisals or business analysis to ensure that the right problem is being solved. The respondents indicated that customer participation is an important factor during COTS software selection as this leads to customer ownership and motivation. Respondents from organisations 6 and 7 recommended using a stakeholder workshops or JAD to elicit and define the user requirements. JAD is said to be an important technique that operationalises user participation and reduces software development costs (Carmel *et al*, 1993).

Identified factors	***Rationale/importance of factors***
Problem definition	• To ensure right problem being solved
Business architecture	• To understand the "big picture" about the business
Business components	• To faciliatate procurement of software components matching business components
Business appraisal/analysis	• To ensure right problem being solved
CASE tools	• To document organisation business architecture
Change in requirements	• To manage changing requirements and new releases of COTS
Cost and time constraints	• To avoid conducting requirements in adhoc manner • To facilitate use of appropriate techniques
Incremental approach	• To address problem of organisational resistance
JAD/ stakeholder workshop	• To operationalise user participation • To reduce software development costs
Object oriented modelling	• To model requirements • To establish important software components in the domain and their interactions
Structured analysis	• To model requirements
Risk analysis (benefit)	• To minimise risks associated with building systems from COTS software • To assess vendor capability and viability
Use-cases and UML	• To document organisation business architecture

Table 6-2. Identified requirements definition factors

The respondents from organisation 5 emphasised the importance of business architecture and that this should drive CBS. They argued that conducting business is becoming increasing complex and therefore it is more difficult for individuals to understand the "big picture" about the business. The respondents proposed using UML and CASE tools to document the organisation's business architecture, and indicated that they found this very useful. Organisations 2 and 5 indicated that high level requirements must be defined in terms of business components. This was supported by organisations 6, 7 and 8 who argued that this would facilitate the procurement or development of software that match these business components.

Respondents from organisations 5 and 7 indicated that they were using both structured analysis and object oriented analysis extensively in the requirements definition stage. However, respondents from organisation 5 were critical of structured analysis and argued that object-oriented analysis is more appropriate for CBS. They suggested an incremental approach to deal with the problem of organisational resistance when migrating to CBS. Similarly, respondents from organisation 7 indicated that identifying business components using object-oriented analysis and design is exceptionally useful. This is consistent with Brown and Short (1997) that object modelling is useful for CBS because it assists in establishing the important software components in the domain and the interactions that occur between them.

There are a number of risks associated with building systems from COTS software that are highlighted in the literature, such as lack of support if the COTS software provider goes out of business (Braun, 1999). Organisation 3 brought out the importance of assessing the risk of using COTS software in a particular context, for example by performing financial checks to assess the vendor stability and viability. Organisations 4 and 8 supported these findings and reported the importance of understanding the market, arguing that selecting a particular product means buying a long-term relationship with a vendor. This corroborates literature that understanding a vendor’s financial stability, track record and long-term strategy is as important as understanding the vendor’s product (SEL, 1996; Boehm and Abts, 1999).

Organisations 6 and 8 brought out the problem of changing requirements, arguing that users change their minds quickly especially when they see a new feature in the new release of one of the products being evaluated. Organisation 2 indicated that qualifying COTS software for high integrity systems is very costly as this may involve 100% requirements testing and auditing the COTS component development process. Organisation 4 reported the problem of time constraints in requirements definition and COTS software selection process. They pointed out that people experienced in requirements analysis and COTS software evaluation are busy with other duties and therefore not available to assist in software evaluation. This corroborates the result of the first study, that lack of time is major constraint to developing software systems (section 5.4.1). Together, this suggests that organisations might be conducting the requirements definition process in ad hoc manner and also not using appropriate techniques in the process.

The identified problems and risks brought out in this study suggest that organisations should adopt risk-mitigation strategies and be willing to invest in CBS in order to minimise on these risks and realise its full benefits. Tran *et al* (1997) propose a number of risk mitigation strategies such as early domain analysis to ensure early establishment of those that are key to the system and development of alternative product integration strategies to ensure rapid replacement of new solutions in response to major changes in requirements. Therefore, procuring and using COTS software products is about managing risks.

The next section presents the important factors identified in this study related to the definition of the evaluation criteria.

6.4.2 Social-technical criteria definition

This section presents the identified factors regarding the definition of the evaluation criteria and are classified into compliance (functionality) issues, quality characteristics, social-economic factors and technology issues.

a. Compliance (functionality) issues

Table 6-3 presents a summary of important compliance (functionality) issues brought out by this field study. All the respondents agreed that COTS software must meet some specific functionality characteristics required by the customer. For example, organisation 3 reported that essential domain specific functionality characteristics of housing rents system for local authorities include rent-setting, rent accounting, rent collection and payment methods, and rent arrears control and recovery. The findings highlight the importance of understanding, deriving and classification of requirements into domain specific functionality characteristics. Tran *et al* (1997) argue that partitioning the system into domain specific subsystems is important because it enables the early identification of candidate COTS products for evaluation.

Furthermore, this suggests that COTS software components cannot be pre-qualified and each project must re-evaluate against some domain specific functionality characteristics that are valid for each project context.

Identified factors	***Rationale/importance of factors***
Functionality (domain specific)	• Assess ability to meet customer requirements • It enables early identification of candidate COTS software
Customer/Organisations standards	• Assess ability to satisfy international standards • To indicate the quality of product
Open system	• Minimise risk when vendor stops supporting product
Organisational policies	• To standardise on products and facilitate sharing data • Help to reduce on products to be evaluated

Table 6-3. Identified compliance factors

Compliance to standards (international or customer) was brought out by organisations 1, 2, 3, 7 and 8 as important criteria, especially in high integrity systems. Organisation 4 indicated that it is their policy that all suppliers must meet the ISO 9000 standards. This is consistent with literature that shows how most organisations insist on ISO 9000 certification of vendors as evidence that vendors use well-defined practices and procedures to produce their products (Haines *et al*, 1997). However, some respondents from organisations 6 cautioned that insisting on compliance to international standards is a problem because they preclude smaller suppliers who may have difficulties with meeting these standards. Therefore, to promote these smaller suppliers, and increase the variety of candidate COTS products, organisations must not be too stringent on these standards.

Respondents from organisations 1, 3, 6 and 8 brought out the importance of organisational policies in COTS software selection (for example, a policy to standardise on Microsoft software products). The inclusion of this criterion helps to screen the available software products and reduce the number of candidate products for further detailed evaluation. However, some organisational policies can be prohibitive and may exclude “better” COTS software from the marketplace. Nevertheless, the advantage with these policies is that they help organisations to standardise on products, which can facilitate sharing of information within the organisation. Therefore, software procurement policies can be useful.

Organisations 1 and 6 emphasised the importance of selecting COTS software that supports open system standards. The respondents were not in favour of selecting COTS software that uses propriety standards or formats but preferred products that makes their data or information accessible with other available commercial tools. The advantage with open systems is that they help to protect products from becoming locked into proprietary solutions especially when a vendor stops supporting the product or goes out of business (Obardnorf, 1997).

b. Product quality characteristics

Table 6-4 presents a summary of important product quality characteristics for the COTS software. For example, organisations 4, 5 and 8 brought out the importance of interoperability when selecting COTS products. Interoperability is the ability of two or more systems or components to exchange information and to use the information that has been exchanged. This was also brought by other organisations, for example organisations 1, 3 and 7 who emphasised the importance of selecting products that can be compatible with (interoperate) Microsoft products.

Identified factors	*Rationale/importance of factors*
Adaptability	• To assess ease of tailoring and extending product
Interoperability	• To assess ability exchange information with other products • To assess ability to integrate with other products
Portability	• To assess ability to be used in other platforms
Reusability	• To evaluate degree to which a product can be used in more than one software system
Scalability and robustness	• To assess ability to work effeciently in different projects
Maintainability	• To assess ease of upgrading and replacing COTS software components
Performance	• To evaluate the compliance of component with specified performance requirements
Reliability & dependability	• To assess fault-tolerance especially in safety critical systems • To certify the product
Efficiency	• To assess use of resources such as memory
Usability	• To assess user friendliness and flexibility of product

Table 6-4. Identified product quality characteristics

The importance of the portability in COTS software selection was also brought out by organisations 1 and 4. They indicated that the evaluation criteria should include assessing the platform in which the product operates, whether it portable between different platforms. ISO/IEC 9126 (1991) defines portability as a set of attributes that bear on the ability of software to be transferred from one environment to another, for example from a Unix operating system to a Windows NT environment.

The importance of the scalability in COTS selection was also brought out by two organisations, that is the ease with which a system or component can be modified to fit a different problem area. Respondents from organisation 4 reported that being a large organisation with a number of projects they prefer to select COTS software that can be used across these projects. In addition, they provided examples of COTS software tools that worked fine on smaller projects but failed to work on larger projects. Similarly, organisation 7 indicated the importance of scalability and robustness arguing that this was the major factor in selecting an Oracle database for their Contracts management system.

The respondents emphasised the importance of selecting products that are reliable and dependable. Reliability is the ability of a system or component to perform its required functions under stated conditions for a specified period of time. The respondent from organisation 3, for example, argued that the systems must maintain full data integrity including restart and recovery features to ensure that no data is lost in the event of a hardware or software malfunction. The most frequently reported product characteristic for the high integrity systems was reliability (organisations 2 and 4). The use of COTS software components for safety-critical systems is a major concern because it requires consideration for fault-tolerance and compliance to certification standards (Beus-Dukic and Wellings, 1998).

Organisations 1, 4 and 6 brought out the importance of efficiency arguing that the COTS product must not use a lot of resources and take a long time to perform important functions. Organisations 2, 7 and 8 reported the importance of including maintainability in COTS selection criteria (i.e., the ease with which a software system can be modified to correct faults, improve performance or other attributes). Respondents from organisations 1, 4, 5 and 7 indicated the importance of selecting products with good graphical user interface (GUI) and suggested that the GUI can assist in assessing the usability. The user interface of the COTS product must be friendly and make it easy for the user to learn and operate the product. Klopping and Bolgiano (1990) argue that to take full advantage of the product capabilities, the product must be easy to use, for example the availability of help functions, appropriate screen prompts and menus, and the flexibility of the software product to accommodate different user proficiencies.

c. Social-economic (non-technical) factors

Table 6-5 provides a summary of identified non-technical factors considered important in the selection criteria. The importance of contractual and legal issues in COTS software evaluation and selection was brought out by organisations 1, 2, 3 and 8. This involves investigating the policy of the supplier's contract and procurement terms, software upgrades and provisions for software fixes within the project schedule. However, contractual issues vary with large organisations normally dictating conditions and terms of agreement. For example, organisation 8 indicated that being a large organisation they do not buy software from smaller companies because of contractual and legal issues. Organisation 2 indicated that they would conduct a financial "health" check on suppliers to assess their viability prior entering into a contract with them.

Identified factors	***Rationale/importance of factors***
Business issues	
Contractual (legal) issues	• To understand vendor's contractual and procurement terms, software upgrades arrangements
Costs and licensing arrangements	• To assess actual cost of product • To help assess cost/benefit of product
Escrow or buy rights	• To assess contigency plans when vendor stops supporting product
Marketplace variables	
Delivery period	• To help in scheduling of projects
Market trends/ changes	• To evaluate vendor/ product reputation • To avoid risk of buying from vendor who runs out of business
Product/technology reputation	• To assess how tested and well-established the product is and potential for support from user-base
Vendor capability	
Local support and training	• To understand useful features and limitations of product
Vendor reputation	• To minimise risk of buying from vendor who runs out of business
Vendor stability	• To benefit from future releases of products

Table 6-5. Identified non-technical factors

The importance of cost issues when selecting COTS software products was brought out by number of organisations. The general cost includes the cost of adapting and integrating the COTS, maintenance (upgrades) costs, training and support costs. The cost of adapting and integrating the COTS is the cost of making changes to the COTS software so that it meets the requirements set for the system. According to organisations 2, 3 and 4 licensing arrangements and product costs are important factors and should be included in the COTS selection criteria. This is the cost involved in obtaining an adequate number of licenses for software development and may include delivery (run-time) version costs, that is the possible costs of obtaining the right to deliver the COTS software as a part of the software to users.

Organisations 1, 4, 5 and 7 reported that what the market thinks about the product determines whether it is selected or not. Organisations 1 and 7 indicated that product reputation in the marketplace is a significant factor because a product with a large user base or that has been used successfully in a certain domain will have a competitive advantage. However a respondent from organisation 2 was hesitant with this kind of approach and argued that it does not necessary follow that if a product is good enough in a civil domain with safety integrity then it can also be reliable in a military domain. This suggests that generalisations cannot be made that the COTS software product will work well because it has been successfully used in one context. Therefore, the product must be re-evaluated when used in a different context.

Respondents considered vendor reputation and capability to be significant factors in the selection of COTS software components (organisation 1, 4, 6 and 8). They argued that it is important to acquire COTS software from a vendor who has financial viability in order to benefit from future releases. The respondents' indication (organisations 3 and 7) that they normally deal with people they can trust and with a good company profile corroborate the results that vendor capability and stability is an important issue when selecting software components. Although the respondents in large organisations (organisations 4, 5 and 7) considered vendor capability and stability to be significant factors in the selection of COTS components, they also indicated that they could always buy rights to the software or buy the company if they were dependent on that software. This is in agreement with Boehm and Abts (1999) regarding the importance of having fallbacks or contingency plans such as product substitution or escrow of failed vendor’s product.

Organisation 6 brought out the importance of the delivery period to assess the vendor's performance and reliability. Organisations 6, 7 and 8 reported the importance of understanding the type of training and local support available from the vendor. The respondents argued that training and support helps users to understand useful features and potential limitations of the software. This is consistent with literature regarding the importance of making agreements with product vendors to provide support for the products and ensure that these products evolve to meet the changing requirements (Tran *et al*, 1997). The findings regarding the identified social-economic factors highlight the importance of considering and including non-technical issues in the COTS software evaluation criteria.

d. Technology factors

Table 6-6 presents a summary of important technology factors for COTS software selection. Respondents from organisations 1, 5, 6 and 7 brought out the importance of evaluating the underlying technology. For example, organisation 1 indicated that COTS component selection should not be based on obsolete technology. Organisation 7 reported that the COTS software product must be compatible with existing technology. The importance of evaluating the COTS software product underlying technology has been highlighted in literature (Brown and Wallnau, 1996a).

Identified factors	***Rationale/importance of factors***
Underlying technology	• To ensure product is not built on obsolete technology
Support for integrability	• To assess capability to inteoperate and integrate with other products
Architectural styles & frameworks	• To avoid product mismatch when integrating with other products
Compatible with existing technology	• To assess support for plug and play
Interface issues	• To assess capability to interoperate with other products • To assess ability to enhance product
Support plug and play	• To assess repaceability and capability to interoperate with other products
Technology standards/ protocols	• To assess support for plug and play • To assess replaceability with other products
Multi-user support, distributed systems and performance	• To assess ability for product support multi-user and operating on distributed network
Availability of documentation	• To understand the interfaces & underlying technology
Security issues	• To assess ability to manage, protect, and distribute sensitive information.

Table 6-6. Identified technology factors

Organisations 3, 5 and 7 brought out the importance of assessing the COTS software technology standards and protocols. Furthermore, respondents indicated that the underlying technology should support plug and play, so that the COTS software can be substituted with a different version or a component with similar functionality from a different vendor. Organisations 2, 4 and 7 indicated that they have started to investigate the technologies that support integrability such as CORBA and DCOM (see also section 3.3.4 for discussion of these technologies). The respondents argued that it is important at an early stage to agree on the underlying technology and the architecture before embarking on selecting the COTS software components. This is important to ensure that the selected COTS software is compatible with underlying technology and to avoid product incompatibilities (mismatches).

Organisation 1 reported the importance of interface issues in COTS software selection. This was supported by organisations 2, 4, 7 and 8. For example, respondents from organisation 7 indicated that they selected MTS technology because it has an interface that allowed them to usefully expose middle tier business rules which encapsulates the database to their suppliers. The interface of a COTS software component can help to add (or hide) functionality of the component, reduce impact of changes to the component and provide the systems integrator with control over the "look and feel" of the component (Vigder and Dean, 1997). Therefore, availability of documentation that describes the COTS software interface is seen as an important factor, as brought out by organisation 1.

Organisations 3 and 4 indicated the importance of having a comprehensive security facility equivalent to, or greater than, BS7799 (that access to the system is via customer identification and password). Organisations 4 and 8 brought out the importance of considering the architectural styles and frameworks of the underlying technology. This is consistent with Tran *et al* (1997) that management of architecture mismatch among the products is needed because each COTS product may vary significantly in functionality and interface making it difficult to integrate them or replace some products when obsolete. The other important technology issues identified in this study include multi-user support, distributed technology and performance of technology. The findings indicate the importance of evaluating both the product and the underlying technology.

The next section presents the important factors related with the identification of candidate COTS software from the marketplace.

6.4.3 Alternatives identification (identifying candidate COTS software)

Table 6-7 shows the important factors associated with identifying candidate COTS software from the marketplace. Respondents from organisations 6 and 8 indicated prior knowledge and past experience as important for identifying COTS software from the marketplace. For example, respondents from organisation 6 pointed out that when selecting COTS software they ask software engineers experienced in that domain to indicate what software they know to be available. Kontio (1995) argues that it is important to utilise the network of people that may have been exposed to the application domain for which candidate COTS are being identified, these include colleagues, experts and consultants. Organisations 6, 7 and 8 brought out the importance of utilising a networking of people and maintaining mailing lists of colleagues in assisting with identifying COTS software products.

Identified factors	***Rationale/importance of factors***
Prior knowledge & experience	• To obtain information about product reputation
Networking and mailing list	• To identify a wide variety of products
Component repository	• Easy to download demo copies and obtain information
Computer fairs and shows	• To view several products and assess market trends • To obtain product information
Internet search	• Easy to download demo copies • To obtain product information
Market surveys	• To obtain product information
Request for proposals/ tenders	• To enable vendors to respond in uniform manner • To allow for transparancy in the evaluation process
Brokerage service	• To identify "best of breed" products
User community	• To identify a wide variety of products • To obtain product information
Vendor promotions & publications	• To see several products and obtain product information
COTS non-availability	• To assist in contigengy planning

Table 6-7. Factors for identifying candidate COTS

Organisations indicated that the Internet search is the most important techniques for identifying COTS components from the marketplace. For example, respondents from organisations 2 and 7 emphasised that the web sites (and repositories) of the big vendors are important sources of information on available COTS software components. The advantage with the Internet search is that most vendors make available demonstration copies of their products on the Internet, which can be easily downloaded and evaluated. Organisations 3 and 8 reported the importance of attending computer fairs and shows to identify several products and obtain product information. This is consistent with Rowley (1993) that attendance at exhibitions, conferences and seminars offers the opportunity to gain an overview of what products are available, the market trends and to examine specific products.

Organisations 7 and 8 brought out the importance of using market surveys to identify candidate COTS software. This involves team members surveying trade journals, vendor literature and the Internet for possible packages. Respondents from organisations 3 and 4 emphasised the importance of using vendor promotions and publications to identify candidate COTS software from the marketplace. This was also supported by organisations 6, 7 and 8. This is consistent with literature that market surveys and vendor literature publications are useful for gathering information about the products (SEL, 1996; Tran *et al*, 1997; Maiden and Ncube, 1998).

It is interesting to note that respondents from this field study (organisations 3, 5, 6 and 8) highlighted the usefulness of public tender procedure or request for proposal (RFP) for identifying software components. The customer can request proposals from the vendors by advertising in trade journals or newspapers. This enables the vendors to describe their packages to the customer in a uniform manner and can help in understanding available packages in the marketplace. The disadvantage with tenders is that some vendors with "superior" products may not respond to the tender leaving the evaluators with inferior products. Tran *et al* (1997) argue that the selection of an inappropriate candidate product for integration can result in an enormous amount of extra time and effort to re-evaluate and re-implement the system with another product.

Respondents highlighted the importance of utilising the "word of mouth" from user community to provide information on available COTS software components in the marketplace. For example, respondents from organisation 3 indicated that professional associations and user community is an invaluable tool for identifying COTS software from the marketplace. Organisations 7 and 8 recommended the use of intermediate brokerage service to identify "best of breed" products from the marketplace.

However, some respondents reported the problem of finding COTS components in the marketplace explaining that this is either because of the non-availability of COTS components or poor marketing strategies by vendors (organisations 2, 3 and 4). The non-availability of COTS software is an important factor in tender procedures, which stipulate that at least three products must be evaluated. Therefore, organisations should be aware of this problem and plan for them.

The next section discusses the important factors, techniques and tools used during the assessment of candidate COTS software against set criteria.

6.4.4 Assessment (evaluation)

The identified factors regarding the assessment (evaluation) are classified into evaluation strategy, data collection techniques and data analysis techniques.

a. Evaluation strategy

In literature, three evaluation strategies have been proposed and these are progressive filtering, keystone identification and puzzle assembly (Oberndorf *et al*, 1997). Organisation 6 indicated that they followed progressive filtering whereby they would start with a candidate set of components. As progressively more discriminating evaluation mechanisms are applied, they can eliminate less "fit" components. Similarly, the internal guidelines document for organisation 4 recommends a progressive filtering strategy for COTS software selection, although the respondents indicated that in most organisations this is not followed because of a lack of time by the software engineers. The disadvantage with progressive filtering is that it is labour intensive (Ncube and Maiden, 1999).

Although organisation 7 were not explicit about their evaluation strategy, analysis of the development of the stores systems project and contracts management systems suggests puzzle assembly approach. The puzzle assembly approach applies an evolutionary prototyping technique to build versions that are progressively closer to the final system. For example, the contracts management systems project involved selection of Oracle database, use of Microsoft Transaction Server (MTS) as middleware and Visual Basic (VB6) for the graphical user interface, in a puzzle assembly style approach. The idea was to link their suppliers through the COM model. However, the project found a lot of difficulties with ensuring Microsoft products (VB6 and MTS) communicate with Oracle databases. This is the problem with the puzzle assembly approach, that software products might not work efficiently with each other.

Organisation 3 reported that they use PRINCE method (CCTA, 1998) on all IT projects, including COTS software evaluation and selection because it provides lower risk to projects and consistent documentation. However, PRINCE is a project management tool and not COTS software evaluation strategy. Organisations 5 and 8 indicated that, prior to selecting the COTS software from the marketplace, it is important to develop the high level architecture and decide on the middleware for integrating the COTS software products. This suggests the use of a keystone approach (see section 3.3.3). The advantage with the keystone approach is that it effectively eliminates a large number of other products from consideration because interoperability with the keystone is the overriding concern. This suggests that a keystone approach might be more useful strategy compared to the other strategies.

The lack of an explicit evaluation strategy in organisations 1 and 2 indicates that the evaluation is most likely conducted in an "ad-hoc" manner. The disadvantage of not using a well-defined evaluation strategy is that inappropriate evaluation methods are used, the process is reinvented each time an evaluation is done and learning from previous cases is difficult (Kontio, 1996). Furthermore, a selection process that is not well defined is vulnerable to organisational politics and pressure. Therefore, to draw maximum benefit from the COTS software selection, organisations should adopt an evaluation strategy and conduct the COTS selection process in a systematic manner.

b. Data collection techniques

Table 6-8 shows the important data collection techniques brought out in this study. Respondents indicated that the user community proved to be a valuable source of evaluation attribute data about COTS software components. This involves gathering subjective opinions and experiences with the products being evaluated by interviewing users or sending questionnaires to them. Therefore, to elicit such data from the user community it may be necessary to subscribe to some mailing lists or keep a directory of relevant contacts phone numbers. It was argued that if the product has been working well with other users it should work well for the new situation in the same application domain.

Identified factors	***Rationale/importance of factors***
Demonstrations	• To understand product features
Experimentation	• To address problem of lack of access to product internals • To assess capability interoperate with other products
Pilot studies	• To undertand product features, faults and ability to interoperate with other products
Qualification of COTS and audit development process	• To understand product features • To certify product for safety-critical systems
Software tests	• To test functionality of products and potential faults
Study documentation	• To understand product features
Template, checklist & questionnaire	• Provide evaluation pattern or model
User community, evaluation	• To obtain opinions and experiences with product • To assess some non-technical issues

Table 6-8. Data collection techniques

Respondents also brought out studying vendor literature as an important technique for collecting attribute data about the COTS products (organisations 1, 4, 6 and 8). This may involve issuing a request for proposals or tender from the vendors regarding some software component and then analysing vendor submissions. Alternatively this may involve studying and analysing product documentation (sales brochures and technical documents) to understand the features available in the products being evaluated. Tran *et al* (1997) argues that reviewing vendor literature can help in the early elimination of inappropriate products, for example rejecting those that do not work with the required operating system. However, organisation 4 indicated the problem of this technique that sometimes documentation is not available from vendors or it is incomplete. Beus-Dukic and Wellings (1998) also suggest that vendor claims must be viewed sceptically.

The problem of how to evaluate COTS software since they are like "black boxes" was brought out by this study and respondents (organisation 1, 3, 4, 5, 6 and 8) suggested using experimentation approach. This corroborates findings from the literature that a comprehensive understanding of COTS software product can only be gained through extensive experimentation (Vigder *et al*, 1996). Experimentation involves hand-on evaluation of the product to assess its compliance with the defined criteria (see section 3.5.3). Experimentation might also involve software testing as reported by organisations 2 and 4. This involves generating test cases and examining the results of the execution of test cases to uncover the symptoms of any faults. Organisation 4 indicated the importance of setting up test cases of the problem the product is expected to solve rather than depend on the vendor's test cases. However, Weyuker (1999) points out that software testing is problematic because it requires a very large number of potential test cases, and therefore it is sometimes hard to determine whether the results of a test case are correct and the repeatability of test results.

Organisations 4 and 7 brought out the importance of conducting pilot studies to evaluate COTS products. A pilot study is an extension of experimentation technique involving “real” data from the organisation to evaluate the product. Therefore, the findings suggest that experimentation with the products in the operating environment in which the product will be used is a useful technique for evaluating COTS products. Other techniques identified in this study for collecting attribute data about the product include auditing the development process, qualification of COTS, use templates and questionnaires. For example organisations 3, 4 and 8 brought out the importance of using templates, checklists and questionnaires. The benefits of templates and questionnaires is that they provide a pattern or model to follow during evaluation and can be useful for preparing product evaluation tests (Maiden and Ncube, 1998).

c. Data analysis techniques

Organisations 3, 4 and 6 indicated that the weighted sum method (WSM) technique is an important technique for data analysis and consolidation of attribute data. For example, respondents from organisation 6 argued that it is important to come up with the scoring matrix and recommended that each element must be ranked according to the “must have” with 10 points and the “nice to have” from 9 to 1 points. The importance of each product is obtained by aggregating all the different points. Similarly, organisation 3 reported that they use a weighting method comprising of mandatory (E) with 20 points, highly desirable (D1) with 10 points, desirable (D2) with 5 points and nice to have (D3) with 3 points. The selection is based on the weighted sum method and the number of times the product passes and fails to meet the mandatory requirements (features). The problem of using WSM has been discussed in literature (see section 3.5.4).

Organisation 3 brought out the importance of feature analysis in COTS software evaluation and selection. In its simplest form, feature analysis provides a list of "yes/no" responses to the existence of a particular property in a product is another form of weighted sum method (Kitchenham, 1996; Kitchenham *et al*, 1997). Feature analysis also provides compound features where the degree of support offered by the product is measured on an ordinal scale. Each feature is accompanied by the degree of importance assessment, for example whether a feature is mandatory or only desirable. The advantage of feature analysis is that it is not restricted to technical evaluations and acquisition issues such as the viability of the supplier can also be evaluated (Kitchenham, 1996; Kitchenham *et al*, 1997). However, as Maiden and Ncube (1998) argue, the problem with feature analysis is that it assumes each product feature is independent of any other which is not the case in most software products. Furthermore, using arithmetic combination to aggregate results from individual feature analysis scores may be misleading.

Many authors recommend using the AHP as an alternative method to the WSM for consolidating COTS software evaluation data (Zviran, 1993; Kontio, 1996; Maiden and Ncube, 1998). The advantages of the AHP technique are that it provides a systematic approach for consolidating information; an objective weighing technique for setting the weighing scale for qualitative and quantitative data, and allows for consistency checking (see section 3.5.4). However, none of the investigated organisations indicated the importance of the AHP or other multi-criteria decision techniques such as outranking method. The results suggest that there are still problems regarding consolidation of evaluation data for COTS software selection. The possible explanation for the preference of the WSM method over the AHP and outranking methods is that WSM is more simple to use (see section 3.5.4). Another explanation is that organisations are unaware of the problems of the WSM.

The next section presents the results of the assessment of the relationship between the important processes supporting COTS software selection.

6.4.5 Relationships between the identified processes

In sections 6.4.1 to 6.4.4 the processes and factors that support COTS software selection were classified into four categories: 1) requirements definition; 2) social-technical criteria definition; 3) alternatives identification; and 4) evaluation or assessment. This section presents the results of the assessment of the relationships between these categories or processes.

a. Effects of requirements definition on other processes

Overall the study indicates that requirements definition influences the other COTS software evaluation processes (see figure 6-1). Respondents from organisations 3, 4, 6 and 8 indicated that the requirements definition has a significant effect on the criteria definition. For example, organisation 3 indicated that initially they write an outline of the basic functional specification of what the system will do. Then the functional specification is converted into a technical specification or criteria. Finally technical specification or criteria is then used as the basis for selection among different vendors. Similarly, organisation 4 supported the findings of organisation 3, indicating that the process of evaluation begins with high level criteria derived from user requirements (e.g. whether the software support a multi-user environment as well as able to communicate with others software). They argued that the high-level user requirements (the reason for wanting a system) must be defined prior to the criteria definition.

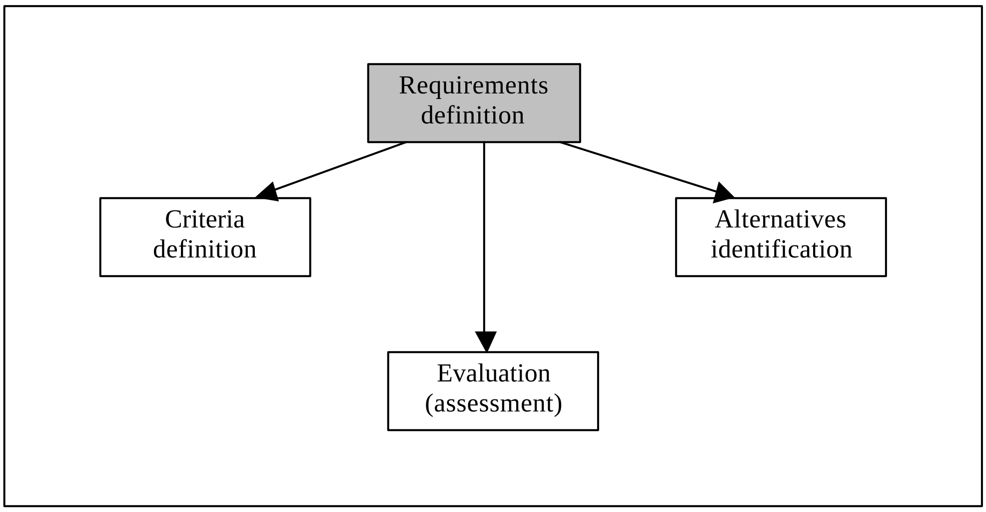


Figure 6-1. Effects of requirements definition process on other processes

Respondents from organisation 6 pointed out that, when evaluating COTS software packages, it is important to prepare an invitation to tender (ITT) document which transforms the requirements definition into a technology type definition. Respondents from organisation 8 argued that business owners and design teams are best placed to consider and decide on software acquisition for their business area by specifying components in terms of business and systems capabilities. The findings suggest that it is important to elicit the high-level requirements prior to the COTS software evaluation process. Further that the high-level requirements must be changed into evaluation criteria. This is consistent with literature that shows how the evaluation criteria definition process essentially decomposes the requirements for the COTS into a hierarchical criteria set (Kontio, 1996).

Organisations 3, 6 and 8 indicated that the identification of candidate COTS software from the marketplace must be driven by some kind of high level requirements. Respondents from organisation 3 argued that good practices for evaluating COTS software require users to first draw up specifications of what they want before embarking on purchasing COTS software packages. Then the evaluation team must search for candidate COTS packages and request suppliers to make presentations highlighting the important features of the products. Organisation 6 indicated that an invitation to tender document (describing requirements) must be prepared, which is then used as a basis for identifying COTS software from the marketplace. Similarly organisation 8 pointed out that the evaluation team specify components in terms of business and systems capabilities and then identify candidate COTS software components from the marketplace. This suggests that the identification of candidate COTS software from the marketplace depends on the requirements definition process.

Organisations 2, 3, 7 and 8 indicated that COTS software evaluation (assessment) must be driven by the requirements definition process. For example, respondents from organisation 2 pointed out that the evaluator must check that the functionality of the COTS software product meets the high level user requirements and that it performs well on the basic test routines (quality attributes). Similarly, organisation 3 reported that users must draw up specifications of what they want, identify candidate COTS software packages and then evaluate the candidate COTS software packages. Organisation 7 stressed the importance of experimenting with the COTS software to test quality attributes and assess how it fits within the organisation's own component model. Organisation 8 supported this and indicated that they started with user requirements and then proceeded to check if the "off the shelf" components would be able to support that component-based development process.

b. Effects of social-technical criteria definition on other processes

Figure 6-2 shows the study findings regarding effects of criteria definition on the other processes. Organisation 6 reported that at times they revise the requirements based on the evaluation criteria and available COTS software products. The respondents indicated that this happens when the evaluated candidate COTS software packages do not meet all the high level requirements but management still want a COTS software solution. The evaluators are asked to revise the requirements, based on available COTS software features (i.e., transforming these attributes into the requirements). This is similar to the experiences of Sledge and Carney (1998) when evaluating COTS products for US department of defence information systems in the domain of human resources and personnel management. Management requested for re-evaluation of products because they were not satisfied with the recommended product and a product that was initially rejected was selected because the products had been substantially improved.

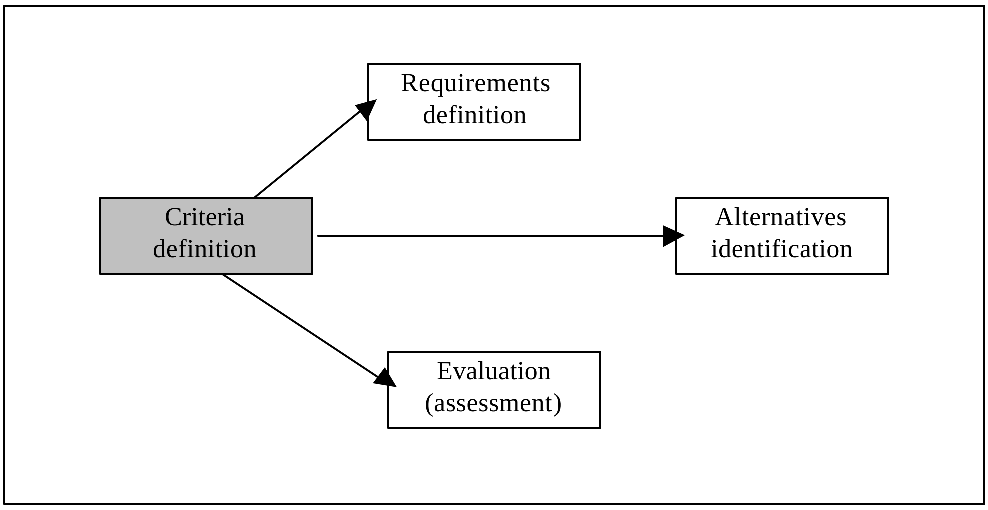


Figure 6-2. Effects of social-technical criteria definition process on other processes

Similarly, organisation 4 indicated that the COTS software evaluation begins with an initial evaluation of the vendors and the products, then the criteria and requirements are refined based on the screened products. This suggests that the requirements will be influenced by the product features and criteria definition. This is consistent with literature that COTS software evaluation is an iterative process between evaluation and requirements definition (Maiden and Ncube, 1998). However, SEL (1996) cautions against this strategy of revising the requirements based on available products and points out that it is important that the evaluation criteria and requirements are not revised in such a way that only one product can be selected.

Organisation 1 and 4 reported that initial search and screening for candidate COTS software depends on the evaluation criteria. Organisation 1 argued that they selected the design tool based on what was considered as industry standard and market viability. Similarly, organisation 4 indicated that the evaluation criteria must be defined first, for example, whether the product supports multi-user processing. Then identification of products in the market based on the evaluation criteria. Organisation 7 also supported these findings that the identification of products from the external marketplace or repository must be based on the defined criteria. They argued, for example, that evaluators must search on functionality, interface type, and technology instantiation. This suggests that the social-technical criteria greatly influence the identification of candidate COTS software products from the marketplace. Therefore when the criteria are not well defined, inappropriate products will be selected.

Respondents (organisation 1, 4, 5, 6 and 8) indicated that the COTS software components are assessed against the social-technical criteria. Organisation 3 indicated that evaluation is not only based on technical criteria but also on other organisational and social factors. For example, vendors are evaluated in terms of services that they provide, their financial stability and the vendor’s understanding of modernising the local council agenda. Organisation 1 indicated importance of assessing the COTS software based on stock market trends, supports for other data formats as well as the proprietary format. Organisations 6 and 7 reported that the evaluation criteria should include functionality, quality attributes, cost, technology it supports, and the organisation's component model. The findings suggest that the use of the social-technical evaluation criteria is important for the selection of COTS software products from the marketplace.

c. Effects of identification of COTS software on other STACE processes

Overall the study indicates that the process of identifying COTS software influences other processes (see figure 6-3). Organisations 5 and 8 indicated that the identification process and available COTS software products greatly influence the requirements. For example, organisation 5 reported that the evaluator must first find out what components are in the marketplace even if they only meet half the requirements because the component can be extended and the requirements revised. Similarly, organisation 8 indicated that they used COTS software components available in the marketplace to understand and define requirements in the organisation. The argument here is that available COTS software in the market place drives the requirements definition. This is consistent with literature that the benefits of COTS software can only be realised with a procurement process that defines requirements according to what is available in the marketplace (Vigder *et al*, 1996).

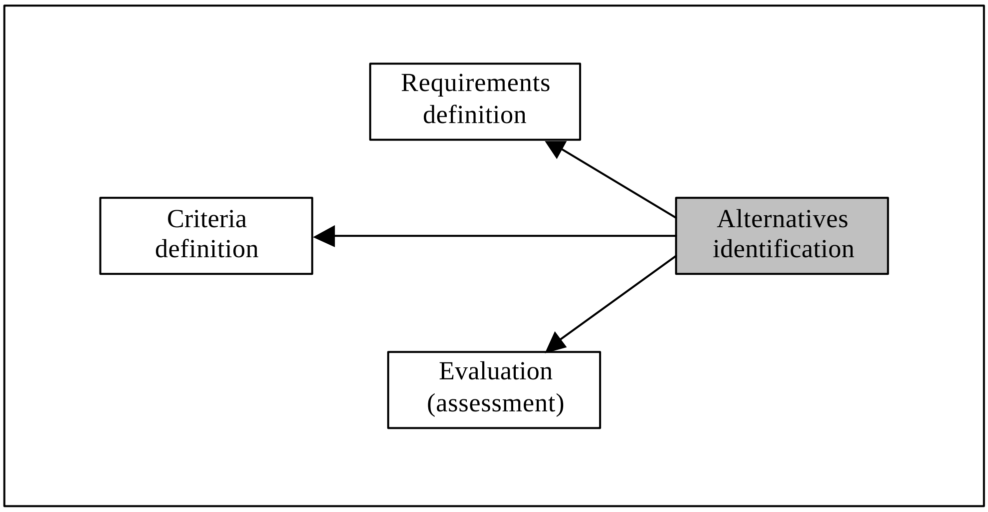


Figure 6-3. Effects of identification of COTS software process on other processes

Organisation 4 indicated that the COTS software product must first be identified from the marketplace and then the identified products initially evaluated to screen for suitable candidate products. Finally, demonstration copies of candidate products are obtained for in-depth evaluation against the criteria. This suggests that the evaluation criteria will be revised based on available COTS software products. This was supported by organisation 8, reporting that they provide business managers with a list of potential COTS software components from the marketplace and their functionality in the form of a checklist. Then managers refine the evaluation criteria and decide whether the products offer them the functionality they require or not. This consistent with literature that initial requirements (and criteria) are revised on the basis of advertisements, package descriptions provided by suppliers, demonstrations, use of packages and comparative studies provided by third parties (e.g., trade papers, etc.) (Finkelstein *et al*, 1996).

Organisation 4 indicated that the evaluation process began by defining the high level criteria and then searching for products in the market that meet the criteria using various techniques, such as market survey, word of mouth from colleagues, Internet search, and publications. Then the selected candidate products are evaluated. This suggests that the identification process and available products in the marketplace influence the evaluation (assessment) of COTS software. Organisations 5, 6 and 8 support this finding and highlighted the importance of allocating human resources and time to identify appropriate COTS products from the marketplace. Tran *et al* (1997) argue that the selection of inappropriate candidate product for integration can result in an enormous amount of extra time and effort to re-evaluate and re-implement the system with another product.

d. Effects of evaluation (assessment) on other STACE processes

The effects of evaluation (assessment) on other STACE processes are presented in figure 6-4. Organisations 4 and 6 indicated that evaluation (assessment) influences the requirements definition. For example, organisation 4 gave an example of the evaluation in which they had begun with a number of requirements for a housing renting system. However, after evaluating some products they observed that one of the benefits of the system was a rent collection feature to help them recover the cost of purchasing the product within six months. This feature became an important requirement for the organisation. Organisation 6 reported that it is important to revise requirements when the evaluated products do not to meet the mandatory requirements, but management still wants a COTS software solution. Furthermore, previous evaluation results in a similar application domain can help to define the requirements and the criteria. This is consistent with literature that storage and management of past evaluation results can help in new evaluation problems particularly in the same application domain (Stamelos *et al*, 2000).

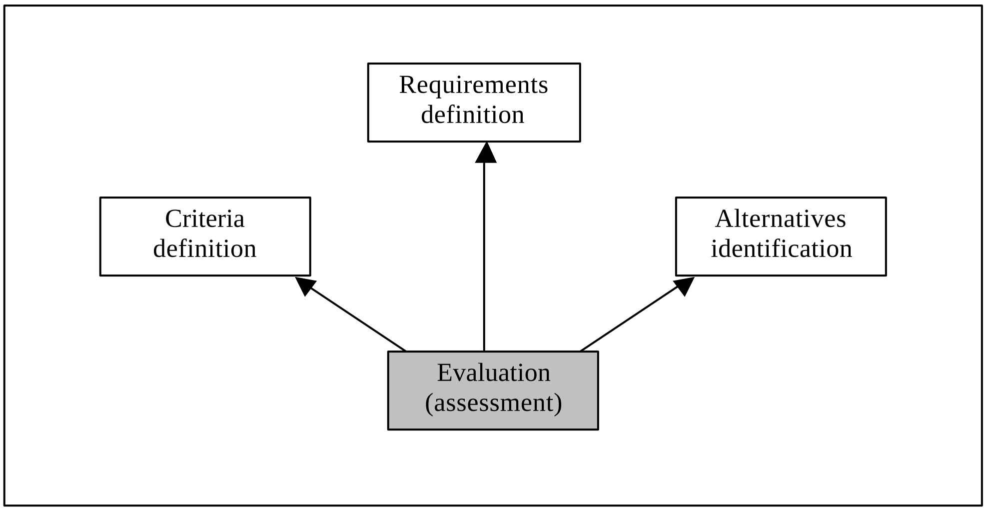


Figure 6-4. Effects of evaluation (assessment) process on other processes

Organisation 4 indicated that the evaluation process begins with identifying the candidate COTS software from the marketplace followed by initial evaluation involving vendor analysis and attending vendor demonstration. The evaluation criteria are then refined and demonstration copies are obtained and evaluated through “hands on” experimentation. Similarly, organisation 6 reported that the evaluation criteria are usually revised when the available COTS software components do not fully satisfy the initial high-level requirements. The findings suggest that evaluation (assessment) can influence social-technical criteria definition. Kontio (1996) argue that organisations that evaluate COTS software frequently benefit from a well-defined, repeatable selection process which facilitates planning, allows for the accumulation of experience, enables a consistent selection process, supports the use of validated methods and increases the efficiency of evaluation.

Organisation 3 and 8 reported that evaluation influences the process of identifying products from marketplace. Organisation 3 pointed out that the evaluation process normally begins with a department showing interest in some particular COTS software product and experimenting with it. However, since tender procedures require that at least three products must be evaluated, the department is advised to identify alternative products. The problem with this procedure is that most likely the department or evaluators will be biased towards the first product initially evaluated.

Organisation 8 indicated that results from previous assessments of suppliers should be used to inform the process of identifying COTS software products. This is consistent with Stamelos *et al's* (2000) finding that it is important to learn from previous evaluations and reuse the results. The problem with this approach is that most likely the information from previous assessments will be outdated.

The next section provides a high level overview of the resulting framework, the Social-Technical Approach for COTS software Evaluation (STACE).

6.5 STACE framework

In sections 6.4.1 to 6.4.4, the important processes supporting COTS software selection were classified into four categories or processes: 1) requirements definition; 2) social-technical criteria definition; 3) alternatives identification; and 4) evaluation or assessment. As a result of the assessment of the relationship between these processes (see section 6.4.5) it was shown that these processes are interrelated. The resulting framework, the STACE can be summarised as shown in figure 6-5. A detailed and full presentation of the STACE framework has been placed in appendix 5.

In the requirements definition process, the high-level customer and systems requirements are discovered through consultation with stakeholders, from system documents, domain knowledge and market studies (Sommerville, 1995). The traditional requirements engineering methods emphasise the technical issues while neglecting the equally important social issues (Jirokta and Goguen, 1994). Therefore, the STACE framework recommends the use of the social-technical approach to systems development (see section 2.3). Customer participation is one of the strategies used in social-technical approaches to incorporate the social issues in the development of the system. In this study the use of JAD or stakeholder workshops was highlighted as an important strategy that operationalise customer participation. Therefore, the STACE framework recommends the use of JAD sessions and review meetings with top management to elicit and validate requirements from stakeholders (see section 2.3.3 for a more detailed discussion of JAD).

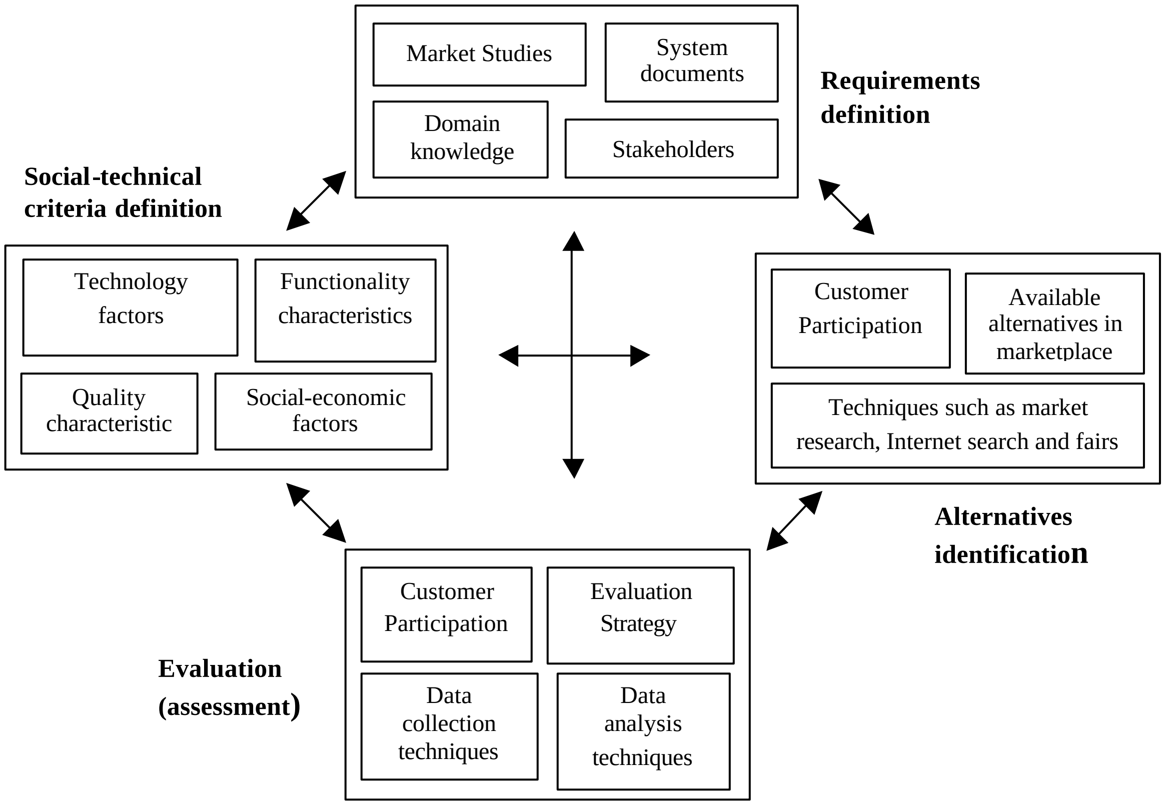


Figure 6-5. **STACE Framework**

In the social-technical criteria definition process, the high-level requirements from the requirements definition phase are decomposed into a hierarchical criteria set and each branch in this hierarchy ends in an evaluation attribute (Kontio, 1996). The STACE framework uses a decomposition approach that is based on social technical analysis (see section 2.3.3) and the AHP criteria decomposition method (see section 3.5.4). The STACE framework recommends decomposition of the high level requirements into a hierarchy of social-technical criteria comprising functionality characteristics, technology factors, product quality characteristics, and social-economic factors (see classification in section 6.4.2). Socio-economic factors are non-technical factors that should be included in the evaluation and selection of COTS components such as costs, business issues, vendor performance and reliability.

The objective of the alternatives identification process is to identify COTS components that meet the high level requirements, so that they can be considered for a more rigorous evaluation (Tran *et al*, 1997). In the STACE framework, this phase begins with identifying the domains relevant to the problem and understanding the types of packages available in those domains. The STACE framework recommends a number of techniques and tools for identifying candidate COTS products. These include networking, mailing list and user community, Internet search, market surveys, invitation to tender (ITT) or request for proposals (RFP), vendor promotions and publications (see section 6.4.3).

The evaluation or assessment phase involves contacting vendor technical support for evaluation information, reviewing vendor documentation and product testing for quality and functionality (Vigder *et al*, 1996). It also includes evaluating COTS performance, interfaces and ease of integration, comparing short-term and long-term licensing costs against integration costs. STACE recommends the keystone selection strategy with the technology as the keystone issue. In the keystone selection strategy, a keystone characteristic such as vendor or type of technology is selected first before selecting the COTS products (Walters, 1995). The separation of COTS underlying technology from COTS products during evaluation allows fair comparisons between products.

The STACE framework also recommends separating the data collection and data analysis of the evaluation. Kontio (1996) argues that the advantage of separating the data collection from analysis is to allow the use of appropriate decision making techniques in the data analysis stage. There are a number of data collection techniques such as examining the products and vendor supplied documentation, vendor analysis, viewing demonstration and interviewing demonstrators, executing test cases and applying the products in pilot projects (see section 6.4.4). STACE proposes selecting appropriate techniques depending on resources and experience. STACE framework recommends the use of the AHP to consolidate evaluation data because of a number of advantages discussed in section 3.5.4.

6.6 Discussion of findings

The major outcome of this field study was that important processes (both traditional and soft factors) that support COTS software components selection were identified and classified (see tables 6-2 to 6-8). The relationships between the important processes supporting COTS software selection was assessed (see figures 6-1 to 6-4). The findings from this study suggest that:

- It is important to develop and agree on the requirements definition and component architecture before embarking on the COTS component selection process. However, some of the organisations indicated that COTS software products in the marketplace must drive the requirements definition. This suggests that while it is important to initially define the high requirements, to realise the benefits of COTS software, a procurement process must be in place that defines requirements according to what is available in the marketplace.
- The stakeholders must be involved in the COTS software evaluation and selection. This is important, first to understand their requirements and secondly, to reduce problems associated with organisational issues such as user resistance. The use of JAD or stakeholder workshop was highlighted as an important strategy that facilitates stakeholder participation because this can also help to reduce development costs and time.
- Most COTS software products do not interoperate with each other, which makes integration of these products difficult. Therefore, selecting the underlying technology for interoperability and testing the COTS product against the adopted technology will assist the organisation in selecting the best technical product. This also suggests that it is important to select the underlying technology prior to selecting the COTS software products. Examples of technologies that support interoperability include CORBA, COM and EJB.
- Selection of COTS software components is not only based on technical factors such as functionality but also on other non-technical attributes. The non-technical issues identified in this study include business issues (contractual and legal issues, costs issues, escrow or buy rights, licensing arrangements), marketplace variables and vendor capability (local support and training, vendor reputation, vendor stability). Therefore, evaluators must remember to include the non-technical factors in the selection criteria.
- The Internet and “word of mouth” from user community are invaluable techniques for identifying COTS components in the marketplace. Other techniques identified in this study include market surveys, tender procedures, vendor promotions and publications. The advantage with the Internet is that most vendors publish their

products on the Internet, which can be downloaded for evaluation purposes. To benefit from the experiences of the user community requires that evaluators maintain good contacts, for example through professional bodies and mailing lists. This suggests that the Internet is a better technique, especially when evaluators are new to that application domain.

- Experimentation with the products in the operating environment in which the product will be used is an effective way of evaluating COTS products. This is because a lack of access to the COTS internals makes it difficult to understand COTS components and therefore compounds evaluation. An alternative to experimentation is to set up a pilot project in which "real" data is used to evaluate the COTS software product. However, the problem with setting up pilot projects is that it is a time consuming and expensive method.
- It is important to use an appropriate data consolidation method. The importance of the weighted sum method was brought out by this study. However, as discussed in section 3.5.4, the problem with the weighted sum method is that it can lead to confusion about the essential requirements and make worst products on important attributes have the highest aggregated scores (Moriso and Tsoukias, 1997; Maiden and Ncube, 1998). This highlights the importance of investigating other multi-criteria decision techniques such as AHP and outranking. The AHP has been recommended in literature as a better technique because it provides for consolidation of both qualitative and quantitative data and allows for consistency checking (Zviran, 1993; Kontio, 1996). Therefore, the usefulness of the AHP for consolidating COTS software evaluation data was investigated further.
- It is important to keep the results of previous evaluations. These previous assessments (evaluations) of COTS software and suppliers can be used to inform future evaluation, for example, the evaluation criteria can be reused for evaluation in the same application domain. This is consistent with literature that it is important to learn from previous evaluation and reuse the results (Kontio, 1996; Stamelos *et al*, 2000).

The identified processes/factors and lessons learnt from this study assisted in the development of the STACE framework presented in section 6.5. The framework provides a classification of techniques and factors that support for COTS software selection. Therefore, the framework will contribute to reducing risks associated with CBS (see section 3.2.3) and the problems of COTS software selection (see section 3.5.1). In particular, the framework incorporated the often-neglected non-technical issues.

The limitation of this study was the possibility of the researcher's subjective bias in analysing and interpreting data. However, efforts were made to reduce on the researcher bias, for example key informants were asked to review the individual organisation study reports. Therefore, this provides the confidence in the validity of the study findings.

Another possible criticism is that the study focussed on 8 organisations from the UK, in 3 instances with only one respondent. However, it is still possible to have a high level of confidence in the research findings and their likely replication in other organisations because the observations held across organisations of different sectors, different sizes and IT characteristics (i.e., identification of similar processes/factors across a wide variety of organisations from different sectors). According to the iterative approach of theoretical sampling adopted in this study, the successive increase in the number of organisations selected to participate should strive for more variations and greater density in data, which would lead to an increase in the generality of the findings. This is similar to choosing a large sample population in statistical studies.

Another reason for the confidence in the study findings is that the respondents and organisations were selected for theoretical relevance (i.e., strong experience in CBS and COTS software selection) until theoretical saturation was achieved. In qualitative research, the concern is not so much the number of people and organisations interviewed (i.e. statistical sampling) but with theoretical saturation (i.e. no new or relevant data seem to emerge)(see section 4.5.5). Furthermore, bringing in documentation from the organisations studied (ie. multiple sources of evidence) widened the research scope and so strengthened the findings.

6.7 Summary

The chapter presented the findings of the second study aimed at identifying important processes and factors that support COTS software selection. A field study comprising a series of interviews in 8 organisations in the UK was used to identify these processes and factors. Explanation building was used to analyse data from this field study.

The chapter presented a number of important processes and factors that support COTS software component selection. The factors were classified into the four processes: requirements definition, social-technical evaluation criteria definition, alternative identification and assessments. The relationship between these processes was also investigated and the results indicate that these processes influence each other. A number of lessons were learnt from this study, for example, that selection of COTS software components is not based only on technical factors such as functionality, but also on other non-technical attributes.

The identification and classification of important processes/factors that support COTS software selection as well as the lessons learnt from this study assisted in developing the STACE framework. The framework needed further evaluation to assess its effectiveness to support COTS software selection for CBS. The next chapter presents the outcome of this evaluation exercise.

7 Evaluating the STACE framework (study 3)

This chapter presents the findings of the third study aimed at evaluating the effectiveness of the STACE framework. The chapter begins by presenting a review of the research method. Then the modification of the STACE workbook for Zambian context is described. Finally, the chapter discusses the findings of the evaluation in Zambia.

7.1 Introduction

The objective of this study was to evaluate the effectiveness of the STACE framework to support COTS software evaluation and selection for CBS (see section 1.4). The STACE framework was developed as a result of the second study and discussed in section 6.5. The evaluation would provide insight into better ways of improving the COTS software evaluation and selection process. The evaluation was undertaken in Zambia a country in which the researcher had experience, as recommended by Fowler (1994) that in cross-border transfer the facilitator must be knowledgeable about both the technology and its target organisation (see section 2.3.4). As discussed in section 4.6, the research questions addressed during this study were:

- How is the framework to be adapted to a developing country (Zambian) context?
- How is the STACE framework rated in terms of gain, interface, quality of life and task support satisfaction?
- How are the special features and principles of the STACE framework rated in terms of usefulness?
- What are limitations (and recommended improvements) to the framework?

The outcome of this study was confirmation of the validity of the STACE framework, showing how a social-technical approach to COTS software evaluation and selection can support CBS.

7.2 Research Method

The research method used for the third study was discussed in section 4.6. Figure 7-1 provides a summary of the main activities undertaken to evaluate the effectiveness of the STACE framework. In the first stage evaluation criteria for the STACE framework were drawn from information systems literature (see section 4.6.2). Then a workbook that operationalises the STACE framework was developed to promote uniformity in the work of participating organisations and reduce threats to internal and external validity of the research findings (see section 4.6.3). A copy of the first version of the workbook is attached in appendix 6[1].

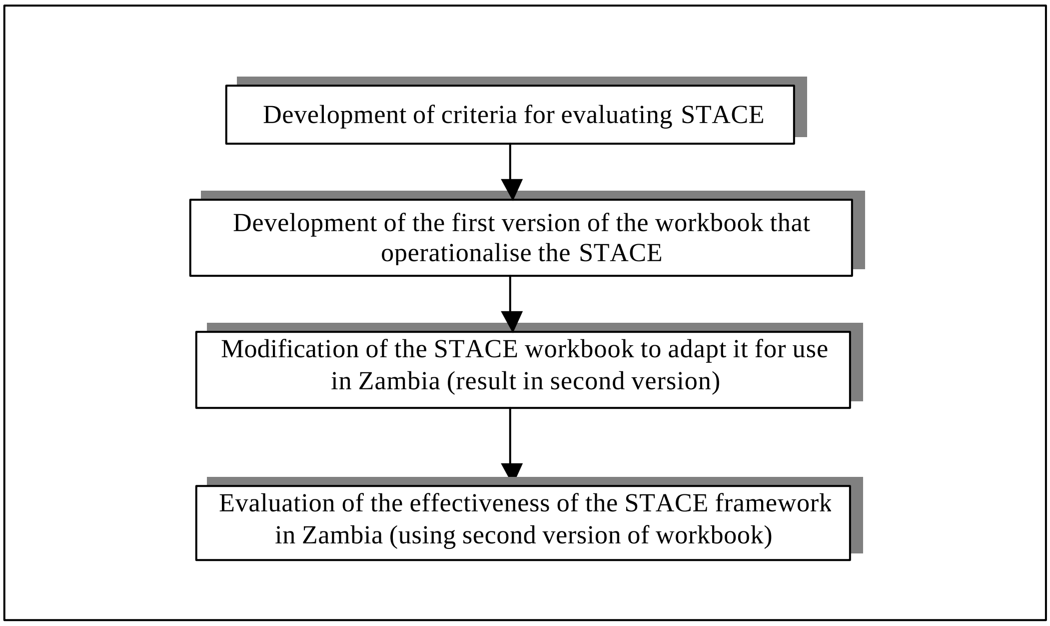


Figure 7-1. Activities for evaluating the STACE framework

The workbook was then modified and adapted to the Zambian context based on literature on information systems in DCs and the findings of study 1, which resulted in the second version of the workbook. The modification of the workbook will be discussed in section 7.4 of this chapter. The framework and second version of the workbook were sent to a number of organisations in Zambia that were perceived to provide rich insights in the problems of COTS software selection. They were requested to apply the framework in their organisations to select COTS software for CBS. This was followed by data collection from the respondents. A multiple case design was adopted for this study, allowing for cross-case analysis and comparison, which is important for identifying patterns.

[1] The first version of the STACE workbook does not include the text underlined in appendix 6.

A data collection protocol was used to guide the researcher in carrying out the case study, which helped to increase the reliability and sharpen the construct validity of the research (see appendix 7). Semi-structured interviews were used to collect data from the respondents (see section 4.5.2 for a discussion of the advantages of the interview technique). Documents were also collected from respondents to corroborate and augment the evidence collected through interviews.

The general mode of analysis used in this study was to identify patterns and provide explanations. This involved first coding the data around: 1) Evaluation criteria categories, to assess the effectiveness of the framework, for example usefulness, ease of use; 2) limitations categories, to identify and classify the problems that organisations experienced with using the framework. Then explanations were formulated to support and validate the identified patterns and case study findings. Data analysis was supported by ATLAS/ti™ a qualitative software analysis tool, which also acted as case study database.

7.3 Modification to the STACE workbook

As discussed in section 2.3.4, many authors argue that techniques and software systems developed with different socio-cultural contexts should be adapted when applied in DCs. For example, Janczewski (1992) identified some technical, economical and cultural problems that impact on IT transfer to West African countries and highlights the importance of adjusting systems when applied in DCs context. The STACE workbook was modified and adapted to the Zambian context using: 1) literature on information systems in developing countries (see section 2.4); and 2) results of the first study aimed at eliciting current CBS practices from Zambia.

7.3.1 Modification of workbook based on literature

The STACE workbook was compared with literature on information systems development in DCs, focussing on the problem areas discussed in section 2.3.3. The objective was to minimise the potential problems associated with technology transfer to DCs. The following modifications were made:

- The worked example included at the end of the workbook to illustrate the COTS software selection (i.e., selecting a wordprocessing software) was improved to clarify the evaluation process. Another example was provided as a PowerPoint presentation on a diskette to illustrate the selection of a database management system (see annex 1 of appendix 6). This aimed to make the workbook easier to use and counter the lack of adequately trained human resources, for developing information systems in DCs such as Zambia (Woherem, 1992a; Corr, 1995).

- Benefits analysis and risk assessments were included in the requirements definition process of the workbook (see annex 2 of appendix 6). This arose from literature which argued that in DCs benefits are overestimated, and discussed from the basis of insufficient insights into the local contexts (Bjorn-Andersen, 1990).
- Vendor's capability to provide training support was included in the evaluation criteria (see annex 2 of appendix 6). This is important because lack of vendor support, lack of education, lack of continued support for further development in DCs has been emphasised in the literature. For example, Bjorn-Andersen (1990) emphasises the importance of including training support for DCs arguing that training in DCs often comes too late, too little and too technical and with the wrong people being trained. Similarly, Avgerou (1990) suggests that the possibility of introducing technology effectively might be significantly enhanced if the workforce at all levels has a high level of education and training.
- The section in the workbook on the Internet search was expanded to provide some suggestions for effective searching of the Internet using public domain tools (see section 4.3 of appendix 6). This is because of potential problems with Internet access in DCs. For example, Petrazzini and Kibati (1999) points out that for most organisations in DCs Internet access is very costly because of lack of infrastructure.
- Advice was provided in the workbook as to which techniques for identifying COTS software from the marketplace would be effective in the Zambian context (see section 4.3 of appendix 6). As discussed in section 2.4.3, economic constraints is one of the major obstacles restricting the application of IT in DCs. Therefore, it is doubtful whether computer fairs/shows, vendor promotions and publications would be effective in Zambia since local software houses do not have the financial capacity to fund them. Therefore, the “word of mouth” from the user community was highlighted as an important technique for identifying COTS software in Zambia because it does not require substantial financial investments.

7.3.2 Modification of workbook based on study 1

The STACE workbook was compared with the findings of the first study aimed at eliciting current CBS practices from Zambia. The objective was to ensure that the CBS techniques that were perceived useful and highlighted by the Zambian organisations in the first study were incorporated in the workbook. The following modifications were made:

- The section on the use of the JAD technique was expanded and reference material was added. This is because the survey conducted in Zambia indicates that customer participation should be encouraged in the requirements elicitation stage (see section 5.5.2). As discussed in section 6.5, the JAD technique is used to operationalise customer participation in the STACE framework.
- The ease of migration to different platforms and conformance to appropriate standards were included in the social-technical criteria (see annex 2 of appendix 6). This is because these factors were considered important by the Zambian organisations for evaluating COTS software product (see section 5.5).
- Experimenting with COTS software was suggested as being the most important technique for evaluating COTS software for Zambia (see section 5.3 of appendix 6). This is because, although studying vendor documentation and attending vendor demonstrations were the most highly rated techniques (see section 5.5.2) and less expensive than other techniques, these techniques pose the risk of selecting products that might not interoperate with other products.
- Templates and references were included in the workbook for those aspects perceived as difficult to understand such as the AHP (see also annex 3 of appendix 6). In addition, a public domain software tool that supports the AHP process was made available. This was aimed at improving the workbook so that it was easy to understand countering the finding from the first study regarding the lack of adequately trained human resources in Zambia (see section 5.5.1).

As a result of these modifications the second version of the workbook was produced (see appendix 6). The next section presents background information about the case study organisations used in the evaluation of the STACE framework.

7.4 Case study organisations

A number of in-depth interviews were conducted with 16 interviewees in 9 organisations in Zambia. These organisations were selected for their experience with CBS and willingness to participate in the case study. Theoretical sampling was used as the basis for inclusion of organisation in the study (see section 4.6.1). Furthermore, a deliberate effort was made to ensure that a wide variety of organisations from different sectors were included in this study (see table 7-1).

Case #	Main business of Organisation	Size of organisation	Interviewee (s)
1.	Retail/Wholesale	Large	Support analyst
2.	Government/ Environment	Medium	Manager and GIS Officer
3.	IT project/ Environment	Small	IT specialist/Consultant
4.	Banking/ Finance	Small	Manager
5.	Public Utilities	Medium	Manager
6.	Research and Training	Large	Lecturers/ researchers
7.	Manufacturing/ Engineering	Medium	Manager
8.	IT Training	Large	System analyst and two lecturers
9.	Government/ Finance	Large	Assistant Director & three Managers

Table 7-1. Case study organisation profile

a. Case 1 (Retail/ Wholesale)

Case 1 is part of an international oil marketing company with over 150 employees at this site. The total company turnover is over £35million. The company has three main departments, finance (auditing, management accounts, IT and financial accounts), human resource, and operations (distribution, commercial, engineering and retail). The organisation has vast experience in the use and application of commercial software in the accounting and finance sector. For example, they are using a Windows accounting software based Oracle database installed in 1995. The organisation used the STACE framework to evaluate accounts and finance software products. The interviewee was from the IT department with over 10 years of working experience and is senior member of staff and reports to the IT manager.

b. Case 2 (Government/ Environment)

Case 2 is a public organisation mandated by the Zambian government to protect the environment and control pollution so as to provide for the health and welfare of persons, animals and the environment in general. The organisation was established in July 1992 with a budget of about £1million through government funding and donors.

The organisation employs over sixty persons, seven of which are in the IT department. The main application of IT in this organisation is in Geographic Information Systems (GIS). A GIS is an information system that is designed to work with data referenced by spatial or geographic co-ordinates (Maguire *et al*, 1991). The organisation has been using ArcInfo 4.2D for DOS and ArcView for Windows for GIS application while some of the users have been trained in AutoCAD. The organisation had just installed a 25-user Windows NT network and used the STACE framework to select new commercial GIS software that operates in the NT multi-user environment. The IT manager and the GIS officer were interviewed. The IT manager is a senior manager reporting to the Director of the organisation and has over 15 years experience in information systems and GIS. The GIS officer is responsible for GIS development within the organisation and has over 5 years of experience in GIS development and implementation.

c. Case 3 (IT Project/ Environment)

Case 3 is a World Bank funded project aimed at promoting Environmental Information Systems (EIS) and networking in Zambia. EIS is the institutional and technical response to the need to improve the role that information plays in environmental management, including both technical and institutional structures through which information is produced and used (Prevost and Gilruth, 1997). The project is designed to establish a forum to facilitate the exchange of information between different agencies and provide information to public and private sector decision-makers. It also aims at developing five district-level issues-oriented information systems focused on key environmental issues. The project commenced in 1998 with a total budget of US$5million for five years. The interviewee was the IT specialist/consultant for the project, a person with over 15 years experience in procurement and development of information systems and commercial GIS.

d. Case 4 (Banking/Finance)

Case 4 is a financial organisation set up to promote foreign investment in Zambia. The organisation assists foreign organisations seeking to invest in Zambia regarding information such as the country investment potential, taxes for foreign organisations and legal issues. The organisation has about 50 employees with an annual turnover of about £5million. The organisation has installed a number of PCs networked through

Windows NT operating system. The organisation applied the STACE framework to selecting some commercial software for monitoring and supporting investments. The manager responsible for the IT department was interviewed. The manager reports directly to the chief executive and has over 10 years of experience in the procurement of commercial software.

e. Case 5 (Public Utilities)

Case 5 is a private utility company providing water and sewerage services to the residents of the City of Lusaka. The organisation's annual turnover is over £5 million. The organisation has installed a number of commercial software applications related to financial systems and utility billing. The organisation has also developed its own billing systems based on a commercial database system (Microsoft Access™). The interviewee was the manager responsible for billing and computing services, who reports directly to the managing director of the organisation. The manager has over 10 years of experience in procurement of commercial software and development of software based on commercial products.

f. Case 6 (Research and Training)

Case 6 is a training institution responsible for training engineering undergraduate in Zambia and has four departments, civil engineering; electrical and electronics; mechanical engineering and surveying. The institution has over 1000 undergraduate being trained in various engineering disciplines and has been supported through donor assistance to maintain the computing facility. The institution has vast experience in the use and application of commercial software for training purposes such as Microsoft Office™, and ArcInfo™ for GIS. Two lecturers responsible for the computer facility were interviewed. The lecturers had about 5 years and 10 years of practical and research experience in computing and software procurement.

g. Case 7 (Manufacturing/ Engineering)

Case 7 is a private company manufacturing and selling soft drinks in Zambia. The company turnover is over £5million. The organisation has seen many changes in terms of ownership and each new owner had their own procurement policies. For example, when the organisation was part of the holding company in the USA it was a requirement that a detailed specification is made and checked by headquarters in

USA, before anything was bought. The organisation has two plants in the City of Lusaka and Kitwe. The Manager for computing services was interviewed. The manager reports to the general manager and has over 20 years of experience in software development, especially on IBM computer systems.

h. Case 8 (IT Training)

Case 8 offers training services in professional accounting and information technology in Zambia and is supported financially by the European Union. The organisation has a turnover of over £50million. The organisation trains about 1,000 students per year. The organisation has been involved in procurement of computer system, network and software systems. Two lecturers in Information Technology and the systems analyst from the IT department were interviewed. The lecturers had over 5 years and 10 years of experience in teaching and consulting in computing and software procurements. The systems analyst had about 5 years experience in developing information systems and procurement of commercial software.

i. Case 9 (Government/ Finance)

Case 9 collects taxes and other revenues on behalf of the Zambian government and has a turnover of over £100million. The organisation also provides advice to government on tax and financial issues. This organisation showed interest in the STACE framework because they were in the process of acquiring a database system. They had narrowed down their choice to SQL Server and Oracle databases. Top management were in favour of Microsoft SQL Server so that they could standardise on Microsoft products while most of the managers were in favour of an Oracle database because they had been trained on Oracle. The final decision was highly political and the organisation did not provide the researcher with specific evaluation results. The Assistant Director and three Managers from the IT department were interviewed to elicit their perceptions of the value of the STACE framework. The Assistant Director has over 15 years of experience while the managers have over 10 years experience in information systems and software development.

The next section presents the results of the cross-case analysis of the evaluation exercise.

7.5 Case study results

This section presents the results of the cross-case analysis of the evaluation of the STACE framework in terms of gain satisfaction, interface satisfaction, quality of life satisfaction and task support satisfaction (see section 4.6.2). The results of the assessment of the usefulness of some features of the framework are also discussed.

7.5.1 STACE framework evaluation

This section presents the results of the cross-case analysis of the evaluation of the STACE framework in terms of gain satisfaction, interface satisfaction, quality of life satisfaction and task support satisfaction (see section 4.6.2).

a. Gain satisfaction

Table 7-2 provides summary results of the evaluation regarding the gain satisfaction. Overall the respondents indicated that they found the STACE framework useful for COTS software selection supporting CBS. The respondent from case 3 indicated that the framework makes the selection process transparent to all stakeholders. Therefore, the framework appears helpful in procurements involving tenders (request for proposals) and when dealing with very complex selection problems. The respondent from case 7 argued that due to its transparency element, the framework would assist with limiting external influences, especially in organisations where people tend to buy from friends instead of reputable companies. Cases 1, 2 and 4 indicated that they were applying some aspects of the framework to their regular work of selecting COTS software.

Criteria	**Results of evaluation/ explanation**
Perceived usefulness	Useful as it makes the selection process transparent to all stakeholders
Decision support satisfaction	It provides for structured decision making process and supports “what if” analysis through the AHP audit trail
Comparison with other guidance – better	It is better because it is systematic and includes non-technical issues
Cost – effectiveness	It is cost-effective because STACE minimises the risk of purchasing from vendors that runs out of business
Clarity – clear and illuminate the process	It clarifies multi-criteria decision problems while some requested for more explanation of AHP.
Appropriateness for task	It is appropriate because STACE is requirements-driven

Table 7-2. Results of evaluation regarding gain satisfaction

Overall, the respondents indicated that the STACE framework does bring about decision support satisfaction. For example, respondents from cases 2 and 3 reported that STACE supports “what if” analysis through the AHP audit trail and provides for a structured decision making process. Furthermore, case 6 pointed out that the framework has the potential to reduce individual bias by involving stakeholders working as a team in the decision making process. However, some respondents from cases 1, 8 and 9 argued that the problem with involving stakeholders is that they are reluctant to participate, they lack time needed for participation and may not even make an effective contribution. This suggests that organisations should adopt strategies that encourage effective customer participation, such as selecting appropriate users and soliciting management support.

Respondents indicated that the STACE framework is clear and did illuminate the COTS software selection process, especially for the IT personnel who understand both technical and non-technical issues. However, some respondents from case 8 sought more explanation on the AHP mathematical background, suggesting that they did not understood some aspects of the AHP. Nevertheless, respondents from cases 5 and 6 argued that the way AHP integrates functionality issues with cost issues did clarify some aspects of multi-criteria decision problems.

Cases 1 and 4 pointed out that the STACE framework is better than other known COTS software selection methods. This was supported by cases 5, 6 and 7 who agreed that the STACE framework is better because it is both systematic and includes non-technical issues. For example, respondents from case 6 pointed out that STACE is useful because most decision-making is done without thorough investigation of the reasons for selecting a particular software. Similarly, case 7 indicated that people often buy from friends without going through an evaluation process. The disadvantages of not using a well-defined process is that inappropriate techniques are used, the process is reinvented for each evaluation and learning from previous cases is limited (Kontio, 1996).

Respondents pointed out that they found the STACE framework appropriate for the task for which it has been developed. For example, respondents from case 2 argued that the framework is appropriate for the task because it allows evaluators to look at business needs or requirements prior to selecting the COTS software. Cases 1, 2, 4 and 6 indicated that the STACE framework is cost effective. This was particularly highlighted in case 1, where a lot of resources were lost because they bought some COTS products from a vendor organisation which collapsed and therefore did not provide the support when they had problems with the products. This suggests that the benefit of selecting the right COTS product outweighs the cost of going through the steps recommended by the STACE framework.

b. Interface satisfaction

The summary results of the evaluation regarding interface satisfaction criteria are presented in table 7-3. Respondents indicated that the STACE framework was perceived as easy to use. For example, cases 1, 3 and 9 indicated that they found it easy to use when using a software tool to support the AHP process. However, case 8 indicated that the framework might not be easy to use for non-technical people. They argued, for example, that the mathematical background material for the AHP is too technical for an ordinary user. In contrast, case 4 argued that the framework was easy to use such that evaluators can easily select the techniques needed depending on the specific COTS software selection context. Case 5 also indicated that they found it easy to use because they have a pool of people with extensive IT and project management experience.

Criteria	Results of evaluation/ explanation
Perceived easy of use	It is easy to use especially with a software tool, but some aspects of AHP may be difficult for non-technical persons
Internally consistent	It is internally consistent because of social-technical criteria is based on requirements and use of software tool
Organisation - well organised	It is well organised but may require structuring to address both simple and more complex COTS software evaluations
Appropriate for audience	It is appropriate for evaluators of COTS software but some users might not make effective contribution
Presentation – readable and useful format	It produces readable and useful format especially the AHP audit trail

Table 7-3. Results of evaluation regarding interface satisfaction

Cases 1 and 4 indicated that the STACE framework was internally consistent. Respondents stressed that the framework was consistent because the social-technical criteria were based on user requirements (first stage) and the criteria were used in the assessment of candidate products (last stage). The use of AHP and a software support tool assisted in maintaining consistency through the entire COTS software evaluation and selection process. Respondents indicated that they found STACE framework well structured and well organised. However, respondents from cases 2 and 6 proposed that the framework should be structured such that evaluators can easily use it for both simple and more complex evaluation cases.

Respondents indicated that the STACE framework was found appropriate for its audience, that is evaluators of COTS software. Whilst respondents from cases 1 and 4 indicated that the framework was appropriate they also stressed that users might not make an effective contribution especially when the evaluation criteria is highly technical. Similarly, respondents from case 2 argued that stakeholders, particularly top management, do not have time for such an elaborate process. Cases 1, 2, 3 and 4 indicated that STACE produces results in a readable and useful format, especially when a software tool is used. Respondents argued that the audit trail produced by the AHP made the whole evaluation transparent, which is very useful when the evaluation results are contested.

c. Quality of life satisfaction

Quality of life satisfaction criteria was formulated to evaluate STACE in terms of user feelings of participation. Cases 1, 4, 6 and 7 indicated that the STACE framework would bring about user's feeling of participation. This was supported by cases 2 and 9 who argued that it brings quality of life satisfaction because customer participation is built within the framework, which would bring about dialogue and consensus with stakeholders.

Although cases 3 and 8 did not explicitly indicate that the framework would bring about user's feeling of participation, they acknowledged the importance of including the non-technical and social issues in the evaluation of COTS software. This suggests the importance of customer participation, since literature indicates that customer participation is an effective strategy for incorporating human and organisational issues such as the design of jobs, work process and usability to bring about quality of work life and job satisfaction (Bravo, 1993; Axtell *et al*, 1997).

While the respondents in cases 1 and 9 pointed out that the STACE framework brought about user feeling of participation, they argued that selecting the right users to participate in the process is important because some users may not make an effective contribution. Further, case 9 indicated that management support is an important factor for securing customer participation. Therefore to benefit from customer participation, organisations require the complimentary task of stakeholder analysis to select appropriate users and the task of soliciting management support to encourage participation.

d. Task support satisfaction

Table 7-4 provides the summary results of the evaluation exercise regarding task support satisfaction. Respondents indicated that the STACE framework produces expected results. However, respondents from cases 2 and 6 qualified their responses by indicating that this is true for complex selection problems in which all the framework guidelines are followed. Further, respondents from case 6 indicated that the framework produces expected results when an evaluation team is used to reduce bias. Respondents from case 4 pointed out that in some cases though they were inclined to think product X was better than product Y, the evaluation results proved otherwise. However, the respondents indicated that they were satisfied with the selection of product Y because the product was selected after a thorough evaluation and consideration of all the relevant attributes.

Criteria	**Result of evaluation/ explanation**
Ability to produce expected results	It produces expected results especially for complex evaluation
Ability to produce relevant results	It produces relevant results especially in tender procurements where transparency is important
Ability to produce usable results	It produces usable results because management has more confidence in results from an evaluation team
Completeness – adequate or sufficient	It is complete but would more usable when analysis and selection of stakeholders for participation is included
Ease of implementation	Easy to implement with software tool but also depends skills and experience of evaluation team
Understandability – simple to understand	It is simple to understand because of the examples provided in the workbook

Table 7-4. Results of evaluation regarding task support satisfaction

Cases 1 and 3 indicated that STACE has the ability to produce relevant results. They argued that the results from the framework are relevant, especially in procurements involving tenders because it provides for a well-defined and transparent process. This was supported by cases 4, 5 and 6. For example, respondents from case 5 pointed out that the framework provides a balanced approach for evaluation of functionality, costs and vendor issues. The findings highlight the importance of using a systematic approach for COTS software selection.

Respondents indicated that the STACE framework produces usable results especially for very complex selection problems. For example, respondents from cases 1 and 2 used the framework to produce results that were used by management in their decision making process. However, respondents from case 6 stressed the importance of using an evaluation team to minimise the problems of bias and subjectivity during the evaluation. This suggests that using an evaluation team can help to produce results that are usable, as management will have more confidence in results produced by a team, than by one individual.

Regarding completeness of the STACE framework, cases 1, 7 and 9 found it adequate and sufficient. This was supported by cases 4 and 6 that the framework was complete and it is up to the organisation to select the appropriate techniques, depending on the evaluation context. However, some respondents from case 8 indicated that the workbook could be improved by reducing the technical jargon and therefore simplified for non-technical users. Another issue that was raised during the evaluation was that some users might not make an effective contribution unless they are properly selected. Therefore, a potentially useful addition to the STACE framework could be the analysis and selection of stakeholder for inclusion in the evaluation process.

Respondents indicated that they found the STACE framework easy to implement. Cases 3 and 9 further argued that ease of implementation is dependent on the major part of the processing done via computer. Similarly, respondents from cases 4, 5 and 8 indicated that they found the Internet and the AHP with support tool easy to implement. However, some respondents from cases 1 and 9 had problems with participation indicating that a number of users were reluctant to participate in the evaluation process. They argued that ease of implementation depends on the skills of the people and experience. Furthermore, case 2 argued that like any learning process they required more practice before they could be confident with using the framework.

Respondents indicated that they found the STACE framework readable and simple to understand. In particular they found the example provided at the end of the workbook very useful. For example, respondents from case 6 were able to select and apply the techniques from the framework as and when required without the researcher's assistance, thus showing that they understood the framework. However, respondents from cases 3 and 8 indicated that some non-technical users might find the AHP difficult in terms of understanding the mathematical theories behind it (see section 7.5.2). Respondents from cases 4 and 6 indicated that the dissemination workshop assisted them in understanding some difficult aspects of framework such as the AHP. This highlights the usefulness of dissemination workshops.

7.5.2 Special features and principles of the STACE framework

The special features and principles of STACE framework were also evaluated in terms of usefulness. The investigated features included the use of customer participation, social-technical criteria, recommended techniques for identifying COTS software from the marketplace and the AHP to support decision-making. Table 7-5 provides a summary of the evaluation results.

STACE special features	Usefulness
Customer participation	• It addresses some organisational and social issues • To satisfy user requirements and facilitate user training • It enhances user satisfaction and acceptance
Use of social-technical criteria	• Successful implementation of software depend on user attitudes and feelings • It clarifies and provides rationale for selecting products based on non-technical issues • It minimises the risk associated with COTS software selection, e.g. vendors that run out of business.
Internet to identify COTS software	• It provides a wide range of choices (both locally and internationally) • Technical support is available on the Internet
AHP to support decision making	• It facilitates the incorporation of non-technical issues • It provides audit trail and is transparent • It reduces the complexity but structuring the software selection

Table 7-5. Evaluation of STACE special features

- **Customer participation**

Overall, the respondents indicated that they found customer participation useful for addressing some organisational and social issues. For example, respondents from cases 1 and 9 indicated that customer participation is important to satisfy user requirements and facilitates user training. Cases 1 and 8 indicated that it enhances user satisfaction and acceptance because users perceive that they are part of the decision making process. The respondents from case 4 further explained that users are very helpful when discussing non-technical issues. Case 8 showed that participation motivates the users to use the selected COTS product, while cases 2 and 8 found that participation is useful for building dialogue and consensus with stakeholders.

However, respondents from cases 1 and 9 argued that although participation is useful, some users do not make effective contribution because they lack the technical knowledge and experience. Furthermore, that the level of knowledge among the users varies making it difficult for some of them to contribute effectively. Case 2 indicated that stakeholders might not have the time to participate in the process. Furthermore, cases 1, 4 and 8 pointed out that it is not easy to involve users because some are reluctant to participate especially when criteria focus too much on technical issues. Therefore, organisations should adopt strategies that encourage participation such as training of users and soliciting management support for participation.

- **Social-technical criteria**

The respondents indicated that that the use of social-technical criteria is useful for COTS software selection. For example, respondents from case 2 argued that successful implementation of software systems depends on the attitudes and feelings of people using them and these people will require upgrades and technical support. Therefore, the social-technical criteria can be seen as an important factor for improving user satisfaction, for example by using a product that has vendor support. The respondent from case 4 indicated that the inclusion of non-technical issues in the evaluation criteria was interesting and would prove very useful for their future work. This is because previously it has been difficult to explain clearly to management why a certain product was recommended for selection when it involved non-technical issues.

However, respondents from cases 8 and 9 pointed out that including non-technical issues in the selection criteria requires additional evaluation costs, which management may not support. In contrast, respondents from cases 1 and 7 argued that the benefits of incorporating the non-technical issues outweigh the additional evaluation costs. They argued that had they used the STACE framework and evaluated the vendor's reputation, they would not have lost resources by buying from un-established vendors. Respondents from case 6 asserted that the social-technical criteria definition is most the significant strength of the STACE framework. They argued that it is important, for example, to include training and re-training in the evaluation criteria when implementing a new system. Therefore, the cost to re-train the people on the new product must be assessed. The findings indicate the importance of including non-technical factors in the selection criteria.

- **Internet for identifying COTS software**

Respondents indicated that the Internet is a useful and easy to use technique for identifying software components from the marketplace. For example, respondents from cases 2 and 4 indicated that the Internet is useful because it provides a wider base for choice and technical support is in most cases available on the Internet. However, respondents from cases 1 and 8 indicated that the Internet is problematic because it produces too much information, making it difficult for the evaluators to know where to start the search. Respondents from case 9 highlighted the security risks associated with buying things on the Internet. Cases 2, 6 and 9 argued that the Internet is expensive in Zambia because of the high Internet Service Provider (ISP) set up and monthly charges as well as high telephone charges. However, case 8 pointed out the benefits of using the Internet to identify products outweigh the costs, for example the Internet can be used to access products available in Zambia and other parts of the world.

The respondent from case 6 indicated that the Internet is not useful for identifying COTS software from the local market (Zambia) because most local software vendors do not publish their products on the Internet due to infrastructure problems and cost issues. Furthermore, the respondent pointed out that because very few people have access to Internet services in Zambia, it means that if the vendor targets the Zambian market and uses the Internet to market their products, they will only reach very few people. They explained that the low Internet access in Zambia is because the Internet has not yet matured and also because of the high cost of access and setting up web space with the local ISPs. Case 3 supported this view citing other problems, such as the lack of facilities for accessing the Internet, i.e., such as computers and modems. Case 9 attributed the problem of low Internet access to the lack of infrastructure such telecommunications network. Therefore, the findings suggest that for the Internet to be viable it must be used in combination with other techniques, such as “word of mouth” from user community.

- **Analytical Hierarchy Process (AHP)**

Respondents indicated that the AHP is useful for synthesising COTS software evaluation data. For example, respondents from cases 1, 3 and 7 argued that AHP is useful because it is scientific and objective. Case 2 indicated that the use of the AHP brought about confidence in the evaluation results and also promoted consensus in evaluation process. Respondents from cases 3 and 4 argued that because the AHP provides an audit trail, it made the whole evaluation process transparent. They argued further that transparency is important when evaluating COTS software products involving tender procedures because the results usually get challenged. Furthermore, respondents from case 4 indicated that AHP is useful for evaluating and consolidating evaluation data involving non-technical issues. Case 6 also pointed out that AHP is useful especially for selecting COTS software for large-scale software systems because it facilitates the structuring of problems into a hierarchy thus reducing complexity. The findings highlight the usefulness of using multi-criteria techniques such as AHP.

However, respondents from case 2 highlighted the problem of AHP in that it involves too many pairwise comparisons when the number of products being evaluated and the criteria increases, which was discouraging to some team members. A software support tool can assist with this problem because it automates the AHP process, performs all the mathematical calculations and performs sensitivity analysis. This was confirmed by cases 8 and 9. Respondents from cases 3 and 8 indicated that some users may want to understand the theoretical background to AHP and this may not be easy for them if they do not have mathematical background. Furthermore, respondents from cases 6 and 9 expressed concerns that some aspects of the AHP are subjective, for example, the priority ranking in the evaluation criteria. They indicated that using an evaluation team as proposed by STACE minimised the risks of bias and subjectivity. This highlights the usefulness of using an evaluation team for COTS software selection.

7.5.3 Problems of the STACE framework

The respondents were also asked to indicate problems they experienced, or potential problems that they might experience, with using the STACE framework to select COTS software supporting CBS. Furthermore, they were requested to provide some recommendations for further improvement of the framework. The main issues that were brought out by the respondents concerning the limitations of the framework include:

- *Lack of time for stakeholder.* Respondents from cases 5, 6 and 9 brought out the problem of lack of time for stakeholders to use the STACE framework. They reported that although STACE is very useful, the decision-makers and users might not have time for such a thorough and elaborate process. Whereas it might take extra time to agree on parameters (criteria) many people would opt for short cuts. Furthermore, respondents from case 2 pointed out that the AHP involves too many pairwise comparisons as the criteria and number of products to evaluate increases. This can discourage stakeholders from participating in the process.
- *Does not adequately handle smaller projects.* Another problem brought out by the study is that the framework does not adequately handle smaller projects or selection of less complex COTS software. The respondents from cases 2 and 6 indicated that the framework is useful for complex problems but more information should be provided on how to select software for smaller projects. Respondents from case 6 suggested that another version of the framework should be developed for selecting less complex COTS software.
- *Can introduce subjectivity and bias in selection.* Cases 2, 7 and 9 indicated that there is a potential subjectivity bias when using the AHP to make pairwise comparisons. They suggested that evaluators might be too subjective, especially when dealing with non-technical issues. The setting up of an evaluation team, comprising representatives from stakeholders, can minimise bias. However, more guidance for STACE users is required on how to deal with this subjectivity bias.

- *Some users might not make effective contribution.* Another problem brought out by cases 1, 4, 7 and 9, was that of users not making effective contribution. They reported that some users might not make effective contribution to COTS software selection process because levels of knowledge in users varies, for example some lacking IT knowledge and experience. They pointed out that users contribution is less effective when the evaluation criteria are weighing too much on the technical side. This has also been brought out in literature that some users from non-technical background are reluctant to participate because they perceive that their contribution might not be useful (Wilson *et al*, 1997).
- *Potential for additional evaluation costs.* Respondents from cases 2 and 8 indicated that the inclusion of non-technical issues and user participation makes the evaluation expensive. They argued that including non-technical issues such as vendor analysis is costly. Furthermore, involving the users in JAD sessions is expensive, first in terms of logistical costs such as venues and also in terms of people's time. The findings suggest that the framework introduce a range of additional evaluation costs.
- *Potential resistance to adopt framework.* The evaluation exercise also brought out the problems associated with adopting the framework. Respondents from cases 2, 4, 5, 7 and 9 brought out the problem of organisational politics in adopting the STACE framework for COTS software evaluation. They argued that decision-makers want to approve what product to select or influence the selection process, sometimes to satisfy external political pressure. Therefore, decision-makers are reluctant to adopt a systematic and transparent evaluation approach for fear of loosing this prerogative. This problem has been identified in literature. For example, Grudin (1994) points out that a system may be resisted if it interferes with subtle and complex social dynamics that are common to group such as threats to political structures.

7.6 Discussion of findings

The most important aspects of the STACE framework evaluation have been the "live" trials and critical comments by organisations from Zambia. This has enhanced the confidence in the validity of the STACE framework for COTS software selection. Overall the findings of this study indicate that the STACE framework was found useful, useable, satisfied user needs and valid for use. However, the study identified some problems and limitations of the framework as a result of its use by organisations from Zambia. The strategies for addressing these problems are discussed below:

1. *Dealing with problem of lack of time.* The study indicated that stakeholders do not have time for such a thorough and elaborate process. To address this problem stakeholders should be made aware of the benefits of using a systematic evaluation method like STACE and motivated to participate in the process. Employing JAD techniques that focus on a meeting (session) delimited by time and set agenda can secure stakeholder participation. Another strategy to secure participation is to solicit management support and to select some user representatives to participate in the evaluation process. Using software tools to support and manage the evaluation process can also help with the problem of time.
2. *Handling smaller projects.* It was brought out that the framework did not adequately handle COTS software selection for smaller projects or organisations. It was suggested that two examples be included in the workbook, one to cater for smaller projects and the other for more complex evaluation problems. This suggestion will be explored so that more guidance is given in the workbook on how to deal with simpler COTS software selection (smaller projects).
3. *Dealing with subjectivity bias.* The study indicated that there is potential for bias and subjectivity from evaluators especially when dealing with non-technical issues. The setting up of an evaluation team comprising of stakeholder representatives and consensus building (as proposed by the framework) can minimise the problem of subjectivity. However, eliminating bias completely in an evaluation is difficult. Care (1978) proposes some conditions that must be met for an evaluation to be considered fair including non-coercion, rationality of participants, joint agreement, disinterestedness, equal and full information to participants. The framework support for audit-trail and transparency helps to eliminate self-interestedness, thus reduces the possibility of bias.
4. *Dealing with users not making an effective contribution.* It was brought out in the study that some users might not make effective contribution in the COTS software evaluation process because the levels of knowledge in users are varied. Selecting appropriate users and training would alleviate this problem. The findings therefore suggest that guidance is required on stakeholder analysis and selection.

5. *Dealing with additional costs.* The study indicated that the inclusion of non-technical issues and user participation makes the evaluation costly. Grudin (1994) argues that demonstrating the system's collective and indirect benefits can help to reduce the disparity in additional work required and benefits. The study brought out the benefits of including non-technical issues in the criteria and the importance of customer participation. For example, respondents argued that customer participation makes training of users easy and is useful for building dialogue and consensus with stakeholders. The respondents also showed that including non-technical issues could minimise the risks of selecting an inappropriate product, for example buying a product from an un-established vendor. Tran *et al* (1997) points out that selection of an inappropriate candidate product for integration can result in an enormous amount of extra time and effort to re-evaluate and re-implement the system with another product. This suggests that the benefits of including non-technical issues and customer participation outweigh the costs of additional work.
6. *Dealing with the problem of acceptance.* The respondents indicated that there could be political resistance to adoption and acceptance of the framework. However, the STACE framework serves as a guide for COTS software selection, and by making the process transparent helps to legitimise the output of the evaluation. Therefore, decision-makers can still use the framework to make selections based on political judgements. Furthermore, as respondents indicated the benefit of the framework is observed in tender procurements or when the evaluation is contested. In addition, the advantages of using a systematic approach such as STACE include the use of appropriate decision making techniques, such as the AHP, to consolidate evaluation data. Therefore, decision-makers should be educated about the benefits of adopting the STACE framework.

A number of important lessons have been learnt from this evaluation exercise regarding COTS software selection. The central findings and the significance of the results on the research are discussed below:

- *Use of a systematic approach to COTS software selection.* Organisational issues such as organisational politics have an influence on the COTS software selection process. Adopting a systematic approach for COTS selection would assist in dealing with these problems. Furthermore the study indicated that a systematic approach would also make it easier to incorporate non-technical issues.

- *Use of social-technical techniques to improve COTS software selection.* The study confirmed that selection of COTS software components is not only based on technical factors such as functionality but also on other non-technical attributes. Therefore, organisations should be willing to include non-technical factors in the selection criteria to reduce the risks and problems associated with COTS software. An evaluation team with representation from all stakeholders should be used, especially for very complex COTS systems in order to avoid bias, encourages dialogue and to build consensus. Customer participation can also help in dealing with some organisational problems in COTS software selection. It is important to solicit for management support for user participation in the COTS software selection process. However, as noted in this study, some users may not make an effective contribution because of low technical competence. Therefore, selection and educating appropriate users is an important activity in the COTS software selection process.
- *Use the Internet and other techniques to identify the COTS software from the marketplace.* Although the Internet is a useful tool for identifying COTS software components from the marketplace, it posed a number of problems. First, some organisations did not have full Internet access because it was very expensive for them. Second, some local software vendors were not yet available on the Internet. Finally, in some cases the Internet produced too much information such that the evaluators were overloaded. Therefore, the Internet must be complimented with other techniques such "word of mouth" from user community. This suggests that maintaining good contacts with the user community through professional bodies and mailing lists can prove useful and helpful.
- *Use a multi-criteria decision technique to consolidate evaluation data.* The AHP proved to be a very useful tool for consolidating evaluation data. The study indicated that AHP promoted consensus in the evaluation process. In addition, because the AHP provides an audit trail, it makes the whole evaluation process transparent. AHP proved helpful to quantify the non-technical issues by making pairwise comparisons and then to synthesise the results. However, AHP involves too many pairwise comparisons when the criteria increases and this may be discouraging to some team members. Therefore, using an appropriate software tool to support the process could help to minimise this problem.

The limitation of this study is that it focussed on Zambia and as discussed in section 2.4.1, DCs differ markedly in terms size of country, nature of industrial structure and development. Therefore, generalisations about the findings cannot easily be made to other DCs with different characteristics, since some of these findings could be tightly related to factors that are peculiar to Zambia or Zambian organisations. Furthermore, the STACE workbook applied in Zambia cannot be generalised for use in other DCs because it was tailored for Zambian context. However, enough insight has been gained from the evaluation exercise about the effectiveness and limitations of the STACE framework as well as the usefulness of the approach adopted to tailor the workbook to Zambian context. The workbook was modified and tailored to the Zambian context using literature on information systems in DCs and the first study aimed at eliciting CBS practices from Zambia. The findings therefore suggest that a similar process would be required to apply the STACE framework in other DCs.

7.7 Summary

The chapter presented the findings of the third study aimed at evaluating the effectiveness of the STACE framework. The framework was then evaluated in nine organisations in Zambia using a multiple-case study approach. Explanation building was used to analyse the data. A number of problems and limitation with the framework were identified, for example that including non-technical issues and participation is costly. The proposed strategies to address these problems were discussed. Overall, the findings suggest that the framework was found useful, useable, satisfied user needs and valid for use.

Research data for this study was predominantly collected using interviews. Therefore, it is possible that the study findings might have been biased by the interview bias. For example, it is possible that the researcher's race may have influenced respondents to give socially admirable but potentially misleading answers. In order to avoid bias efforts were made to ask open-ended questions in a neutral way and then respondents were asked to explain their responses. The responses that were not supported with an explanation were disregarded in the analysis. Furthermore, efforts were made to collect documents from the organisations invoking triangulation of research data, thus encouraging a convergent line of enquiry. This provides confidence in the validity of the findings of the third study aimed at evaluating the STACE framework.

There is potential for the application of these results from both a research and a practical perspective. From the research perspective, it is advancing the state of knowledge about the social-technical approach to COTS software evaluation and selection. The primary significance of this work is the existence of a framework for applying social-technical approaches to COTS software evaluation and selection. From the practical perspectives the STACE framework and workbook provide guidance to practitioners regarding COTS software evaluation and selection. The contributions for practitioners and for research are discussed in more details in the next chapter.

8 Summary and conclusion

This chapter reviews the research question and method used to answer the research question. It then summarises the research findings, discusses the research contribution and highlights the research limitations. Finally, directions for future research are suggested and the conclusions of the thesis are presented.

8.1 Research question and method

Building information systems is not an easy task. In the context of developing countries this is made worse by the constraints imposed by the economic and social climate. In order to facilitate this process, the overall goal of this research as encapsulated in the research question (see section 1.4), was set as:

> ***What processes (including traditional and soft factors) provide support for evaluating and selecting software components for COTS-based systems?***

In order to answer this research question, three immediate objectives were formulated. The first was to understand how CBS can provide support for organisations, by studying the potential benefits and risks associated with CBS as well as by eliciting current CBS practices. Secondly, to identify important processes (including traditional and soft factors) that support COTS software component selection in CBS. Lastly, to provide a generic social-technical framework for COTS software evaluation and selection that supports CBS. To achieve these objectives, the overall research process was split into three main studies, as summarised in table 8-1.

The first study was aimed at eliciting and synthesising current practices and potential benefits of CBS from the UK and Zambia. A survey approach was adopted for this study through administration of self-completion questionnaires. A survey approach was adopted because of the descriptive purpose of this study. Systematic sampling was used in this survey so that statistical inferences can be made about the population from responses to the sample. The sample comprised of 36 respondents from UK SMEs, 59 respondents from UK software houses and 20 respondents from Zambia.

Simple statistical analysis was used to illuminate the results from this survey. The *mean* and *standard deviation* were used as a standard to compare (identify similarities) and determine the relative importance of variables. The Scheffe test was used to assess the differences between the UK and Zambian samples.

	Study 1	**Study 2**	**Study 3**
Objective	Elicit and synthesise current practices and potential benefits of CBS (also facilitate the identification of problems and solution with CBS)	Identify processes and factors that support COTS software selection (also develop framework for COTS software selection)	Evaluate the effectiveness of the STACE framework (also facilitate the adaptation and refinement of the framework)
Research questions	What are the current practices, process and techniques for building systems using COTS software? What are the benefits, costs and risks associated with CBS? What kinds of problems (and solutions) related to CBS development? What are the difference between a developing country (Zambia) and the UK in terms of CBS?	What are the most important processes and factors that support COTS software evaluation and selection? How can these processes and factors be classified (and how do they relate to each other)? How can the social-technical approach used to improve the COTS software selection? What kinds of problems (and solutions) have organisations experienced in evaluating COTS components for CBS?	How is the framework rated in terms of gain, interface, quality of life and task support satisfaction? How is the framework to be adapted to developing country context? What are limitations (and recommended improvements) of the framework? How are the special features and principles of the STACE framework rated in terms of usefulness?
Research Method	Survey	Interviews and paper documentation	Case study
Outcomes	Better understanding of current CBS situation and focussed research direction	Development of the STACE framework	Confirmation of the validity the STACE framework

Table 8-1. Organisation of the research

The second study was aimed at identifying important processes (including traditional and soft factors) that support COTS software component selection for CBS. A series of interviews allowed cross-organisation comparisons, which is important for identifying patterns and developing theoretical categories. The predominant method used to collect data was face-to-face interviews. However, documents were also collected to corroborate and augment this evidence. To increase the reliability and enhance the construct validy of this study, an interview protocol was used to guide data collection procedures. A total of 8 organisations from the UK participated in the study. Explanation building was used to analyse data from this study, which involved the categorising, tabulating, explaining and modelling of data. To increase construct validity key informants reviewed and provided feedback on the draft reports.

The third study was aimed at evaluating the effectiveness of the STACE framework to evaluate COTS software supporting CBS. A workbook that operationalises the STACE framework was developed to promote uniformity in the work of participating organisations and reduce threats to internal and external validity of the research findings. A multiple-case study approach was used and 9 organisations from Zambia participated in the evaluation exercise. Interviews and documentation were used to collect data from the respondents. The general mode of analysis used in this study was to identify patterns and provide explanations. This involved first coding the data around: 1) Evaluation criteria categories, to assess the effectiveness of the framework, for example usefulness, ease of use; 2) limitations categories, to identify and classify the problems that organisations experienced with using the framework. Then explanations were formulated to support and validate the identified patterns and case study findings. Data analysis was supported by ATLAS/ti™ a qualitative software analysis tool, which also acted as case study database.

8.2 Summary of the research findings

This section presents the research findings from the literature review, study 1 (eliciting current CBS practices), study 2 (identifying important processes/factors) and study 3 (evaluating STACE framework).

8.2.1 Literature review

Understanding the current state of literature is important because it provides the background and starting point in any research. In the literature review, both academic and professional literature on information systems and social-technical approaches, information systems in DCs, building systems from COTS software, evaluation and selection of COTS software and other relevant topics were reviewed. This identified the research gap and established a theoretical background in information systems and software engineering domains. The literature on information systems highlighted the problems of information system development and implementation including organisational and social issues. The use of social-technical approaches in addressing these organisational and social issues was identified as beneficial.

The literature on information systems in DCs identified problems such as the lack of adequately trained human resources, economic constraints, lack of infrastructure, socio-cultural issues and applications problems. The literature review on CBS highlighted a number of potential benefits that DCs could tap into including reducing software development and maintenance costs. Therefore, initial research focussed on investigating how CBS can be used to support information systems development in DCs. However, as a result of the first study and further literature review, a number of problems and risks associated with CBS were identified, including COTS software evaluation and selection. This led to refining the focus of the research to COTS software evaluation and selection.

8.2.2 Findings of study 1 (eliciting current CBS practices)

The overall goal of the first study was to elicit and synthesise current practices and potential benefits of CBS from the UK and Zambia. Overall a number of similarities between Zambia and the UK can be seen. For example, both indicated that the main obstacle to developing software systems is the lack of adequately trained human resources. This confirms the importance of adopting strategies, such as CBS, for developing software that have the potential to reduce development costs and ease of use. Both the UK and Zambia brought out that reducing software development costs is the most significant benefit of CBS. Building systems from COTS software is cheaper because the essential requirements need not be specified in detail (as with bespoke systems) and the cost of COTS component is shared among a number of users.

The survey shows that the most significant technology used by practitioners for COTS integration in the UK and Zambia is Microsoft's OLE and DDE. However, OLE and DDE are limited to Windows operating systems and are not portable to other platforms. This suggests that there are problems regarding CBS and therefore organisations should be encouraged to invest their resources in other component integration technologies such as CORBA and Enterprise JavaBeans (see section 3.3.4). Regarding the requirements engineering phase, the most frequently used techniques in both the UK and Zambia are observation, prototyping and demonstrations suggesting that organisations focus on requirements elicitation rather than on modelling.

The most significant technique for evaluating COTS software brought out by UK software houses and Zambia was studying documentation. However, the problem with relying on studying documentation is that the selected product might not be compatible with other existing software. Therefore, the findings suggest that there are problems with the COTS software selection process. Furthermore, the low rating by the three samples of systematic approaches to COTS software selection such as multi-criteria decision-making techniques further augment the suggestion that COTS software evaluation and selection is problematic both in the UK and Zambia.

The survey also brought out significant differences between Zambia and the UK samples, for example regarding the problem of lack of time in building software systems. Organisations in the UK consider time as an important resource in software development. Furthermore, the results indicate that there are differences between the UK and Zambia regarding organisational factors (i.e., organisation structure and politics, changing business strategy, organisational resources and support, and external factors). This suggests that there are social-cultural and contextual differences. Therefore, the findings highlight the problems of transferring systems and frameworks developed in the UK to Zambia.

Based on the findings, there are differences between Zambia and the UK regarding using inventory of existing COTS software technique for identifying COTS software from the marketplace. Further, there were differences of opinion between what to include in the evaluation criteria (i.e., the price, ease of migration and conformance to appropriate standards). This is consistent with literature that evaluators sometimes include immaterial and inappropriate attributes in the criteria leading to incompatibilities (Kontio, 1996). Therefore, the differences between Zambia and the UK regarding the evaluation criteria and the techniques for identifying COTS software indicated some potential problems with COTS software evaluation and selection requiring further investigation.

The major outcome of this study is a documented and better understanding of the current situation, problems (and solutions) people have experienced in relation to CBS. The similarities and differences between the UK and Zambia brought out a number of problems associated with building systems from COTS software. In particular, the problem of COTS software evaluation and selection to support the CBS process was highlighted. This resulted in a more focussed direction for the overall research project. Therefore, the second study was aimed at identifying important processes/factors that support COTS software evaluation and selection. The results from this survey also assisted in later adapting the framework developed from the UK data for use in Zambia.

8.2.3 Findings of study 2 (identifying processes/factors for COTS selection)

The overall goal of the second study was to identify important processes (including traditional and soft factors) that support COTS software selection for CBS from the UK. The findings from this study suggest that it is important to develop and agree on the requirements definition and component architecture before embarking on the COTS software selection process. Further, that while it is important to initially define the high level requirements, to realise the benefits of COTS software, a procurement process must be in place that defines requirements according to what is available in the marketplace. The study also indicates that it is important to involve stakeholders in the COTS software evaluation and selection in order to understand their requirements and to reduce problems associated with organisational issues such as user resistance. The use of JAD was highlighted as an important strategy that facilitates stakeholder participation and that it can also help to reduce development costs and time.

The results of this study suggest that it is important to select the underlying technology (e.g., CORBA, COM and EJB) prior to selecting the COTS software products. This is because most COTS software products do not interoperate with each other, which make integration of these products difficult. Therefore, selecting the underlying technology for interoperability and testing the COTS product against the adopted technology will assist the organisation in selecting the best technical product. The study also indicates that the Internet and "word of mouth" from user community are invaluable techniques for identifying COTS components from the marketplace.

The advantage with the Internet is that most vendors publish their products on the Internet, which can be downloaded for evaluation purposes. To benefit from the experiences of the user community requires that evaluators maintain good contacts, for example through professional bodies and mailing lists.

The findings indicate that selection of COTS software components is not only based on technical factors such as functionality but also on other non-technical attributes. The non-technical issues identified in this study include business issues (contractual and legal issues, costs issues, escrow, licensing arrangements), marketplace variables and vendor capability (local support and training, vendor reputation, vendor stability). Therefore, the study suggests that evaluators should incorporate non-technical issues in the selection process. Customer participation is an important technique that addresses organisational and social issues, for example it is argued in literature that customer participation would encourage system ownership and motivation.

The study indicated that a lack of access to the COTS internals makes it difficult to understand COTS components making the choices difficult. Experimentation with the products in the operating environment in which the product will be used was brought out as an effective way of evaluating COTS products. However, there is the need for some way to put all the information together in order to be able to make a reasoned and clear decision that can be justified with a clear audit trail. The importance of investigating multi-criteria decision techniques such as AHP and outranking was highlighted in this study because of the problem with the weighted sum method. The study focussed on the AHP because it provides for aggregation of both qualitative and quantitative data and allows for consistency checking (Zviran, 1993; Kontio, 1996).

The major outcome of this study was that the important processes and factors that support COTS software components selection were also identified and classified. The identification and classification of these processes/factors as well as the lessons learnt from this study assisted in the development of the framework (i.e., STACE) for COTS software selection. The STACE framework was evaluated in the third study and will be reviewed in section 8.3 of this chapter.

8.2.4 Findings of study 3 (evaluating the STACE framework)

The aim of the third study was to evaluate the effectiveness of the STACE framework for selecting COTS software supporting CBS. Overall, the findings of this study suggest that the STACE framework was found useful, easy to use, simple to understand, valid for use, and to bring about user and quality of life satisfaction. A number of important lessons have been learnt from this evaluation exercise regarding COTS software selection. For example, the results of the study suggest that it is important to use a systematic approach, such as STACE, for COTS software selection to address organisational and non-technical issues. Furthermore, the study highlighted the importance of using an evaluation team with representation from all stakeholders to mimimise subjective bias, encourage dialogue and to build consensus.

In this study, the AHP proved to be a very useful tool for consolidating evaluation data. The respondents perceived that AHP promoted consensus in the evaluation process. In addition, because the AHP provides an audit trail, it makes the whole evaluation process transparent. AHP proved helpful to quantify the non-technical issues by making pairwise comparisons and then to synthesise the results. However, the study indicated that the AHP involves too many pairwise comparisons when the criteria increases and this was discouraging to some team members. The findings therefore suggest the importance of developing an appropriate software tool to support the process and minimise these problems.

The evaluation exercise indicated that although the Internet is a useful tool for identifying COTS software components from the marketplace, it posed a number of problems. For example, some organisations did not have full Internet access; some local software vendors were not available on the Internet and it was perceived by some organisations that the Internet produced too much information. The results suggest that the Internet should be complimented with other techniques such as "word of mouth" from the user community.

The general lessons from this evaluation exercise were extracted to establish good practice and learning for current and future research. However, the study also identified some limitations to the framework including problems over the lack of time; handling smaller projects; dealing with subjectivity bias; users not making an effective contribution; additional costs. A number of strategies for addressing these problems were suggested, the major strategy being the use of a software tool to support the AHP and manage the evaluation process. Other strategies proposed include educating decision-makers about the benefits of adopting the STACE framework; selecting appropriate users to participate in the evaluation process and soliciting management support for participation.

8.3 Review of the STACE framework

This section reviews the objective of STACE framework, compares to other frameworks and discusses the applicability of the framework.

8.3.1 Objective and principles of STACE

As discussed in section 3.3.1, COTS software evaluation and selection is problematic for a number of reasons. These include the lack of a well-defined process, the "black box" nature of COTS software, rapid changes in the marketplace, misuse of data consolidation methods and neglect of non-technical issues. STACE is a generic framework for COTS software evaluation and selection aimed at addressing these problems through a social-technical approach. It provides guidance for the organisation regarding the process of evaluating and selecting COTS software components for CBS (see appendix 6). The important principles and features of the STACE framework include (see appendix 5):

- To provide support for a systematic approach to COTS evaluation and selection.
- To provide support for evaluation of both COTS products and the underlying technology. It proposes a keystone evaluation strategy in which the underlying technology is selected before selecting the COTS products.
- The use of social-technical techniques to improve the COTS software selection process. It recommends the use of a social-technical evaluation criteria and customer participation in the COTS selection process.
- The use of multi-criteria decision-making techniques to consolidate evaluation attribute data. It proposes that this could be effectively achueved through the use of the AHP.

8.3.2 Comparing STACE with other frameworks

Table 8-2 presents a summary of the comparison of STACE with other generic COTS software evaluation and selection frameworks, which were discussed in section 3.4.

Framework	Characteristics	Strengths	Weaknesses
SSEF	• It proposes a top-down approach that identifies the important elements that a software system must include to foster high-level understanding. • Uses knowledge world's concepts (i.e., usage world, development world and system world). • Multiple viewpoints approach to evaluation (user satisfaction and economic returns). • Defines the elements (dimensions, factors, and categories) clearly to facilitate evaluation and reduce the evaluators' conflicting viewpoints. • It is organised along three dimensions corresponding to the software's producers, operators, and users.	• It provides a baseline for establishing metrics programs in organisation (Boloix and Robillard, 1995). • It offers a broad system snapshot by considering a number of different perspectives (end users, developers, and operators) (Brown and Wallnau, 1996b). • A top-down approach has the advantage of flexibility, permitting extensions by following a predefined pattern (Boloix and Robillard, 1995).	• It is not specific to COTS selection and the issues of how to define the evaluation criteria are not addressed (Kontio, 1996). • It gives little detailed insight into the strengths and weaknesses of a technology in comparison with its peers (Brown and Wallnau, 1996b).
OTSO	• Provides explicit definitions of tasks in the selection process, including entry and exit criteria; (Kontio, 1996) • Advocates incremental, hierarchical and detailed definition of evaluation criteria; • Provides a model for comparing the costs and value associated with each alternative, making them comparable with each other; • Uses appropriate decision-making methods to analyse and summarise evaluation results.	• It addresses the complexity of COTS software evaluation (Brown and Wallnau, 1996b). • The systematic repeatable process can promote learning through experience and improve the COTS selection process (Kontio, 1996). • The use of the AHP provides evaluation consistency (Zviran, 1993) and provide structured information (Kontio, 1996).	• AHP is only appropriate when there are few comparisons and when all criteria are independent (Maiden and Ncube, 1998) • Neglect of non-technical issues or "soft" factors (Powell *et al*, 1997)
Delta	• Evaluate a new software technology by examining its features in relation to its peers and competitors • It is a systematic approach that includes modelling and experiments. • That technology evaluation depends on understanding technology "delta" descriptions of how a new technology's features differ from other technologies. • Evaluates how these "delta" differences address the needs of specific usage contexts.	• It provides techniques for evaluating the product underlying technology. • It can also facilitates individual product evaluations that concentrate on their distinguishing characteristics in relation to their technology precursors and product peers (Brown and Wallnau, 1996a).	• It focuses on technology evaluation and neglect product and vendor evaluation • It does not address the political and economic factors that often separate a winning technology from other contenders (CMU, 1998).
PORE	• It integrates existing requirements engineering methods and other techniques such as feature analysis and multi-criteria decision-making. • It is template-based (templates provide guidelines for conducting evaluation). • It advocates for a parallel and an iterative requirements acquisition and product selection/rejection.	• It provides guidance to model requirements for COTS software selection • The parallel requirements acquisition and COTS software selection means requirements acquisition informs COTS software selection and vice versa.	• Use of traditional approaches make it vulnerable to neglect of social issues • It is labour intensive (Ncube and Maiden, 1999).
STACE	• It supports a systematic approach to COTS evaluation and selection • It proposes a keystone evaluation strategy in which the underlying technology is selected before selecting the COTS products. • It uses social-technical techniques (i.e., social-technical criteria and participation) to improve the COTS software selection process. • It uses multi-criteria decision-making techniques (i.e. AHP) to consolidate evaluation attribute data.	• It addresses the non-technical issues through use of social-technical techniques. • It supports evaluation of both COTS products and the underlying technology. • It provides for reuse of lessons learnt from previous evaluation cases by maintaining a database of evaluation results. • Use of the AHP promotes consensus, transparency and consistency checking.	• It does not adequately deal with evaluation of software for smaller organisations (projects). • It increases the cost of the evaluation process because of inclusion of non-technical issues and user participation. • Some aspects of AHP having subjective bias. • Some users may not make an effective contribution.

Table 8-2. Comparing STACE with other COTS software selection frameworks

The SSEF focuses on assessing the software product, process and their impact on the organisation but it does not provide guidance on how to define the evaluation criteria. While the OTSO addresses the complexity of COTS software component selection and provides a decision framework that supports multi-criteria component selection analysis, its weakness is that it neglects the non-technical issues or “soft” factors. The Delta technology framework is useful for evaluating new tachnology but it does not address the political and economic factors that often separate a winning technology from other contenders. PORE is a template-based method to support requirements acquisition for COTS software products. The weakness of PORE method is that it is labour-intensive and vulnerable to the neglect of social issues.

In general what is missing in these frameworks is how to address the “soft” issues or the non-technical factors, such as costs, organisational issues, vendor capability and reputation. Therefore, STACE was developed to facilitate a systematic requirements-driven COTS software selection and address this problem using social-technical techniques. Furthermore, STACE supports the evaluation of both COTS products and the underlying technology while the other frameworks in table 8-2 either emphasise product or technology evaluation. The use of the AHP in STACE promotes consensus, transparency and consistency checking. Another advantage of STACE is that it provides for reuse of lessons learnt from previous evaluation cases by maintaining a database of evaluation results.

However, as with all theoretical models, the STACE framework has some limitations including that of not adequately dealing smaller organisations (projects); increasing the cost of the evaluation process; that some aspects of AHP may have a subjective bias; and some users not making an effective contribution (see section 7.5.3). Notwithstanding the limitations of the STACE framework, it was found useful for evaluating selecting COTS software. Furthermore, a number of strategies were recommended to address these limitations (see section 7.6). For example, it was suggested that using an appropriate software tool to support the evaluation process could help to minimise problems associated with the AHP. Demonstrating the framework's collective and indirect benefits can help to reduce the disparity in the additional work required (i.e., increase in cost) and benefits. Selecting appropriate users and training would alleviate the problem of users not making an effective contribution in the COTS software selection process.

8.3.3 Applicability of the STACE framework in DCs

Alford (1994) points out that successful technology transfer requires the researcher to show the immediate benefit of the technology being transferred. Some aspects of the STACE framework and method were being used by three of the organisations that participated in the evaluation exercise. Another organisation requested that a web site for the STACE framework should be available so that others who did not have an opportunity to participate in the evaluation can download the documentation. It was also suggested that the Computer Society of Zambia should be approached to adopt the STACE framework guidelines for COTS software selection. This suggests that the framework was found useful by organisations in Zambia and therefore could be successfully transferred.

However, the STACE framework was evaluated in only one developing country (Zambia) with the help of a workbook that was tailored to the Zambian context. Therefore, it cannot be directly generalised to other DCs because of the diversity of these countries in terms of size of country, endowments of resources, nature of their industrial structure and levels of per capita national incomes. To apply STACE framework in Zambia, the workbook was modified and tailored to the Zambian context using (a) literature on information systems in DCs and (b) the first study aimed at eliciting CBS practices from Zambia. As discussed in section 7.6, to apply the STACE framework in other developing countries requires tailoring the workbook to that specific developing country context. Therefore, a similar process to the one used for Zambia would be required to modify the STACE workbook for a different DC (see section 7.3 for details).

8.4 Study contribution

This research has a number of implications for both researchers and practitioners, particularly those involved in selecting COTS software to support the CBS process. This section discusses the contribution of this research to each of these perspectives.

8.4.1 Academic contribution

The primary significance of this work is the development of a generic social-technical framework for COTS software evaluation and selection (i.e. the STACE)(see section 8.3). The STACE framework provides a classification of important processes (including traditional and soft factors) that support COTS software selection. The framework also allows the classification of a set of techniques and tools within each process. It highlights relationships between processes (and factors within each process) and thus facilitates the examination of relationships between factors in different processes and their impact on CBS success. Therefore, the framework could be used for research purposes not only in COTS software evaluation research but also in the wider information systems and software engineering research field.

The STACE framework addresses the weaknesses of the existing evaluation models by attempting to ensure that all issues (including non-technical) are structured and addressed. This is achieved by integrating social-technical criteria as well as customer participation in the COTS software evaluation process. Customer participation provides for consensus building during evaluation by allowing evaluators and stakeholders to discuss and agree on evaluation parameters. The framework also provides a structured evaluation model; thus allowing the designation of a hierarchy of selection criteria based on organisational needs. The evaluation and selection process is clearly defined, in terms of processes, techniques to be used and activities to be performed.

The study also provides a deeper understanding of CBS practices, social-technical approaches and of information systems development in DCs. It highlights the impact of organisational and social issues on information systems success, and how social-technical approaches can address them. Following a characterisation of the problems of developing information systems in DCs, it identified current strategies adopted by DCs and the potential benefits of CBS for DCs. Furthermore, the study provides an analysis of the risks associated with CBS and in particular COTS software evaluation problems. In addition, the study brought out the need to focus effort on non-technical issues when evaluating COTS software.

8.4.2 Contribution for Practitioners

The research findings provide software engineers and information systems professionals with a systematic requirements-driven framework that they can use to plan and implement COTS software evaluation and selection for CBS. The framework incorporates the often-neglected non-technical issues. It provides a classification of the important factors, techniques and tools that practitioners can use to support COTS software evaluation and selection.

The STACE framework applies the social-technical techniques (i.e., a social-technical criteria and customer participation) to COTS software evaluation. This research demonstrates that the use of a social-technical criteria and customer participation plays an important role in improving the COTS software evaluation process outcomes. Customer participation is regarded as an effective strategy for incorporating human and organisational issues in the evaluation process. It leads to perceived ownership and acceptance of the systems, thus increasing the chances of successful implementation. The use of social-technical criteria ensures that the non-technical issues related to COTS software evaluation are incorporated. Therefore, the framework contributes to reducing risks associated with CBS and COTS software selection.

Furthermore, the STACE framework also recommends use of multi-criteria decision-making (MCDM) techniques to consolidate evaluation attribute data. MCDM techniques can help decision-makers to rank the alternatives in decreasing order of preference and therefore help in selecting the “best” products among alternative COTS software products. This research shows that using a MCDM technique such as AHP promotes consensus in the evaluation process, makes the evaluation process transparent and provides a means of measuring the consistency of the decision-maker's judgements.

A number of recommendations have emerged from this research to help practitioners to successfully evaluate and select COTS software. These include:

- Sufficient time should be allowed for COTS software evaluation. This will facilitate the use of a systematic approach and the selection of appropriate techniques for each phase of the selection process. Nevertheless, organisations should be careful that the cost of evaluation does not exceed the benefits of the COTS software being procured.

- Include of non-technical issues in the evaluation criteria. Because selection of COTS software is based on both technical and non-technical issues, therefore, practitioners should use requirements techniques that allow the elicitation of non-technical and social issues, e.g. JAD.
- To encourage customer participation, evaluators should solicit management for support. This is because customers or users are reluctant to participate in CBS and COTS software selection. Management can persuade users to participate, for example by providing incentives to them when they do participate.
- The use of the Internet to identify COTS software from the marketplace should be supported by other techniques such as "word of mouth" from user community. This is because some organisations might not have access to the Internet because of financial and infrastructural constraints. Furthermore, some local products may not be available on the Internet. Therefore, maintaining contacts with the user community through professional bodies and mailing lists can be useful and helpful.
- To prevent selection of products that are not compatible with existing technology, practitioners must experiment with COTS software products within the operating environment prior to selection. Relying on vendor literature and user community alone is inadequate and it poses a risk of selecting products that might work well on their own but not interoperate with products.
- Ensure the COTS software evaluation and selection process is transparent. This is especially important for selections involving tenders (request for proposals). Using a systematic approach such as the STACE framework supported by a multi-criteria decision technique, to consolidate evaluation data, can help to make the process transparent.

8.5 Limitation of the study

Most of the limitations of this research project are related to the limitations of the method used to collect and analyse data. In the first study, a self-administered survey approach was used to collect data from respondents. The major criticisms against self-administered survey approaches include low response rates, little control over events under study (Robson, 1993; Sapsford, 1999). In this study a number of strategies were used to deal with these problems, as discussed in section 4.4.2.

The major limitation of the field study method used in the second study was that it depended on few organisations (i.e., small sample) thus potentially limiting the scope of study. However, the iterative approach of theoretical sampling adopted in the field study ensured that the successive increase in the number of organisations selected to participate would provide more variations and greater density in data, which would lead to an increase in the generality of the findings. Moreover, bringing in documentation from the organisations studied (ie. multiple sources of evidence) widened the research scope and so strengthened the findings.

The limitation of the case study method in the third study was the possibility of the researcher bias. For example, interviews were used to collect data from the research sites and therefore there is potential for interview bias, that is the researcher unwittingly influencing the responses. In order to avoid bias efforts were made to ask open-ended questions in a neutral way and then respondents were asked to explain their responses. The responses that were not supported with an explanation were disregarded in the analysis. However, it is difficult to completely remove the problem of interview bias. For example, it is possible that the researcher's race and gender may have influenced respondents to give socially admirable but potentially misleading answers because they wanted to please the researcher. Therefore, multiple sources of evidence were used to counteract this problem and encourage convergent lines of enquiry.

Another problem related to researcher bias is the researcher's subjective bias in analysing and interpreting data. Explanation building was used to analyse data from the case studies. Efforts were made to reduce the researcher's bias during the analysis of case data; for example informants were allowed to review the case study reports. Furthermore, a software tool was used to support the data analysis process and also act as a case study database. The case study database provides a chain of evidence so that an external observer can follow the derivation of any evidence from case study data to the case study conclusion.

The rigorous nature of the data collection and data analysis methods employed in this research therefore provides evidence of the validity of the research findings. The findings in this research are based on studies undertaken in the UK and Zambia (developing country). Therefore, generalisation about the findings cannot be made to other DCs with different characteristics, since some of these findings could be tightly related to factors that are peculiar to organisations studied or to Zambian organisations. However, as discussed in section 8.3.3 the research findings can be relevant to other DCs, for example the STACE workbook can be tailored and modified for other DCs.

8.6 Future research

The identification of important processes and factors supporting COTS software evaluation and selection has highlighted a number of areas that require further research. For example, future work can focus on the examination of each of the identified factor and its impact on the COTS software selection process. Further work can also investigate the relationships between the factors and the organisations, or draw conclusions about COTS component selection in different organisations.

One of the criticisms of the STACE framework was that it did not efficiently handle smaller projects. It was argued that the framework did not provide enough guidance on how to select COTS software for smaller projects. This is an important criticism in the DC context because of the prevailing economic constraints. Therefore, future work could include characterising problems associated with using each technique proposed in the framework. The framework would then provide a classification of factors, techniques and tools according to cost and financial considerations. This would enable organisations in DCs to select the appropriate techniques according to their financial capabilities.

The evaluation of the STACE framework revealed the problem of additional costs introduced by the inclusion of non-technical issues in the evaluation criteria and customer participation. Furthermore, that the evaluation process takes a longer time because of additional work. A number of templates were provided to speed up the process and reduce the additional work of reinventing for evaluation criteria, each time the evaluation is done. However, the problem was not completely solved. Therefore, future work can focus on the development of a software tool to support all the processes in the STACE framework. The software tool would automate the evaluation process; suggest techniques and criteria according to the type of evaluation problem; management of past evaluation results in order to inform future evaluation cases; and support the use of a multi-criteria decision method. This would initially involve developing the prototype and then testing it in a number of organisations.

In the STACE framework, the AHP was proposed for consolidation of evaluation data. The AHP was found useful in that it incorporates both objective and subjective measures into decision making process. However, it presented a number of problems, for example that AHP has a potential for bias and subjectivity especially when dealing with non-technical issues. Furthermore, that the AHP was time consuming because of the mathematical calculations and the number of pairwise comparisons that increased as the number of alternatives and criteria increases. Therefore, further work can focus on developing a software tool that supports the AHP and addresses some of these problems. Future work can also investigate the use of other multi-criteria techniques, such as outranking.

This research focussed on COTS software evaluation and selection supporting the CBS process. However, this research has implications for the other stages of the CBS and information systems for DCs. For example, systems built using COTS packages will require maintenance and enhancements, some prompted by vendor updates and changing customer requirements. Therefore, determining procedures or guidelines for deciding when to accept upgrades would be an interesting research area. In this study, lack of skilled human resources has been identified as the principle barrier to developing software systems in DCs. Another interesting areas of research therefore is investigating COTS software integration skills and its impact on the CBS success in DCs.

8.7 Overall Conclusion

Although there have been many advances in the information technology (IT) field, most DCs have not yet fully benefited because of the specific problems experienced by these countries. COTS-based systems offers a number of benefits that the developing country can tap into, such as reducing development and maintenance costs and improving reuse across projects. COTS-based systems depend on successful evaluation and selection of COTS software to fit customer requirements. There are a number of problems associated with COTS software selection including lack of well-defined evaluation process, “black box” nature of COTS software, rapid changes in the marketplace and neglect of non-technical issues. Therefore this research focused on COTS software evaluation and selection supporting COTS-based systems.

The central research question was “what processes (including traditional and soft factors) provide support for evaluating and selecting software components for COTS-based systems?” The answer to the research question lies in a generic social-technical framework for COTS software evaluation and selection (i.e. STACE), which provides a classification of important processes, factors, techniques and tools for COTS software selection. The framework can also be used by software engineers and information systems professionals to plan and conduct COTS software selection for COTS-based systems. While not a complete answer to the multi-faceted problems faced by information systems in DCs, the STACE framework does go some way to help guide the direction for both future theoretical work, as well as being a practical tool, useable in real contexts.

9 Appendix 1: List of publications

The results presented in Chapters 5, 6 and 7 have been presented/ published at a number of selected conferences and journals. These publications are listed below:

1. Kunda D. and Brooks L., "Applying social-technical approach for COTS selection," Proceedings of 4th UKAIS Conference, University of York, McGraw Hill, pp. 552-565, April 1999.

2. Kunda D. and Brooks L., "Case study: Identifying factors that support COTS component selection," Requirenautics Quarterly, RESG of the British Computer Society, Vol. 17, pp. 7-9, June 1999.

3. Kunda D. and Brooks L., "Case study: Identifying factors that support COTS component selection," Workshop on COTS, International Conference on Software Engineering (ICSE99), Los Angeles, May 1999.

4. Kunda D. and Brooks L., "Identifying processes that support COTS-Based Systems in Developing Countries," Proceedings of doctoral Consortium on Advanced Information Systems Engineering (CAiSE99), Heidelberg, pp. 128-139, 1999.

5. Kunda D. and Brooks L., "Assessing success of a social-technical method for COTS software selection: A survey approach," Proceedings of second International Conference on Enterprise Information Systems (ICEIS 2000), Stafford, pp. 294-298, 2000.

6. Kunda D. and Brooks L., "Identifying and Classifying Processes (traditional and soft factors) that Support COTS Component Selection: A Case Study," Proceedings of ECIS Conference, Vienna University of Economics and Business Administration, Vol. 1, pp. 173-180, 2000.

7. Kunda D. and Brooks L., "Assessing organisational obstacles to component-based development: A case study approach," Information and Software Technology Journal Elsevier Science, Vol. 42, pp 715–725, 2000.

8. Kunda D. and Brooks L., "Identifying and Classifying Processes (traditional and soft factors) that Support COTS Component Selection: A Case Study," European Journal of Information Systems, Vol. 9, No. 4, pp. 226-234, 2000.

9. Kunda D. and Brooks L., "Assessing important factors that support Component-Based Development in Developing Countries," Information Systems for Development Journal, 2000.

10 Appendix 2: Cover letter and reminder card for study 1

Dear Respondent

BUILDING SYSTEMS FROM COMMERCIAL-OFF-THE-SHELF (COTS) SOFTWARE

The Department of Computer Science at the University of York is conducting a survey of some aspects of current industry for building systems from Commercial-Off-The-Shelf (COTS) software. The study aims to elicit information about current practices of organisations and to enable us to identify best practices. For the purpose of this research COTS software are defined as:

> *Commercial Off-The-Shelf (COTS) software is a component, subsystem or system, sold or traded to the general public usually by third-party vendors at prices based on established catalogue or market prices. The source code for these components is not available to the buyer nor does the buyer control specification, release schedule and evolution of the components. Examples include word processors, operating systems, libraries and functional modules.*

Enclosed is a copy of the questionnaire, which takes about 20 minutes to complete. It is recommended that someone responsible for specifying, procuring and developing software systems should complete this questionnaire. After completion please return the questionnaire in the enclosed self-addressed envelope. Your organisation's response will be kept strictly confidential. No data will be associated with individual respondents or organisations and the results will only be analysed and reported in aggregate form.

We appreciate your help in our research effort, therefore if you would like a copy of our completed study please indicate this on the last page of the questionnaire. The report of this survey will be available in June 1998 and we will make certain that you receive a copy of our results. We believe that you will find the questionnaire both interesting and provocative and look forward to receiving your reply.

Thank you in advance for your time and co-operation.

Yours faithfully,

Douglas Kunda Dr. Laurence Brooks

Research Student Research

QUESTIONNAIRE REMINDER CARD

16th June 1998

Dear Sir/ Madam,

Four weeks ago I sent you a questionnaire to complete. The questionnaire was aimed at eliciting current practices of organisations in building systems from Commercial-Off-The-Shelf (COTS) software. If you have already replied please accept my most sincere thanks. If not, I would greatly appreciate it, if you could complete the survey and return the questionnaires in the stamped addressed envelope provided.

Even if you do not build systems from COTS software please still return the questionnaire indicating this. It is important that your response is included in the study, which aims to benefit the IS professional community and organisations which plan to, or build systems from COTS software components.

If by some mistake you did not receive the questionnaire or have mislaid it, please do not hesitate to get in touch with me in 01904 432737 or via e-mail

Supervisor

11 Appendix 3: Questionnaire for study 1

A SURVEY OF CURRENT PRACTICES FOR BUILDING SYSTEMS FROM COMMERCIAL –OFF-THE-SHELF (COTS) SOFTWARE.

The purpose of this questionnaire is to investigate and elicit understanding of the current situation, problems (and solutions) people have experienced in relation to building systems using Commercial-Off-The-Shelf (COTS) software. Commercial Off-The-Shelf (COTS) software is a component, subsystem or system, sold or traded to the general public usually by third-party vendors at prices based on established catalogue or market prices. The source code for these components is not available to the buyer nor does the buyer control specification, release schedule and evolution of the components. The questionnaire is divided in four sections as follows:

Section I. Demographic Information. This section collects information about the survey respondents, their company and general information about COTS.

Section II. *Overview of Development Process*. This section examines current practices about the development process applicable when building systems from COTS software. It also examines the costs, benefits and risks associated with COTS software use.

Section III. *Requirements Engineering Phase*. This section examines current practices, techniques and tools used during the requirements engineering phase

---

SECTION I. DEMOGRAPHIC INFORMATION

A. RESPONDENTS DETAILS

1. Respondent's First Name: ____________________ Surname: ________________________________

   Postal Address: ____
   _
   ____ _______________

   City/Town: ______________________________ Postal Code: ____________________________

   Email address: ____________________________ WWW address: __________________________

Telephone: ______________________________ Fax: ______________________________

2. Please check the category that best describes your main job function in your organisation
   - ❑ Management
   - ❑ Systems analysis or design
   - ❑ Application or systems programming
   - ❑ Hardware Specification
   - ❑ Operations
   - ❑ Academic or Research
   - ❑ Procurement
   - ❑ Other (*Please specify*) _____________

3. How long is your work experience with building systems from COTS components: ________ years

B. COMPANY DETAILS

1. Name of your company:
2. Please check the category that best describes the primary business activity of your company.
   - ❑ Banking/ Finance
   - ❑ Insurance
   - ❑ Manufacturing
   - ❑ Construction/ Engineering
   - ❑ Retail/ Wholesale
   - ❑ Mining
   - ❑ IT services
   - ❑ Public Utilities
   - ❑ Government
   - ❑ Other (*please specify*) _______________

3. Please indicate your company's annual turnover?
   - ❑ Below £1million
   - ❑ £1million - £5million
   - ❑ £5million - £35million
   - ❑ £35million - £100million
   - ❑ £100million - £500million
   - ❑ £500million - £1billion
   - ❑ Above £1billion
   - ❑ I don't know
4. Please indicate the number of employees in your company
   - ❑ Below 10
   - ❑ 10 – 50
   - ❑ 50 – 250
   - ❑ Above 250

5. What have been the main constraints or obstacles to developing software systems (both bespoke and COTS) have you experienced? Indicate your strength of agreement for each statement below.

	Strongly Disagree				Strongly Agree
a. Lack of financial resources	1	2	3	4	5
b. Lack of adequate trained human	1	2	3	4	5
c. Lack of time or too tight schedules	1	2	3	4	5
d. Lack of Institutional framework to support IT development (e.g. there is no	1	2	3	4	5
e. High development and maintenance	1	2	3	4	5
f. Political and social issues in	1	2	3	4	5
g. External environmental factors (e.g. economic setting or government actions)	1	2	3	4	5

h. Other (*please specify*):

SECTION II. OVERVIEW OF DEVELOPMENT PROCESS

A. COST, BENEFITS AND RISKS ASSOCIATED WITH SYSTEMS BUILT FROM COTS SOFTWARE

1. What do you consider as the benefits (gains and improvements) of building systems using COTS? Indicate your strength of agreement for each statement below.

	Strongly Disagree				Strongly Agree
a. Reduces software development costs	1	2	3	4	5
b. Reduces the expense of maintaining	1	2	3	4	5
c. Improves reuse across projects	1	2	3	4	5
d. Cost effective obsolescence	1	2	3	4	5
e. Promotes competitive marketplace enabling system integrators wide range of	1	2	3	4	5

f. Other benefits (*please specify*): ___________________________________

2. What do you consider as the major barriers (difficulties) and risks associated with using COTS software? Indicate your strength of agreement for each statement below.

	Strongly Disagree				Strongly Agree
a. Lack of acquisition guidelines	1	2	3	4	5
b. Difficult to discover the actual technical capabilities of COTS software	1	2	3	4	5
c. Instability due to periodic releases of	1	2	3	4	5
d. Loss of schedule control during development and maintenance	1	2	3	4	5
e. Legal implications in case of system	1	2	3	4	5
f. Difficult to identify and resolve product incompatibilities (mismatches)	1	2	3	4	5
g. Additional functionality causes side	1	2	3	4	5
h. Additional qualification certification	1	2	3	4	5
i. Failure to meet requirements	1	2	3	4	5
j. Lack of support if COTS provider goes out of business	1	2	3	4	5
k. Difficult to select from vast array of	1	2	3	4	5

l. Other risks (*please specify*): ____________________________________

B. GENERAL INFORMATION ABOUT COTS

1. Do you generally use external software development houses to build systems from COTS software?

❑ N ❑ Yes

2. Please check the box(s) that describe the approaches that you are using to build systems from COTS software components. Select ALL applicable.

- ❑ We purchase COTS software components and use them without adapting or extending them.
- ❑ We purchase COTS software components and then adapt or extend them for our local needs.
- ❑ We purchase COTS software components and integrate them into software systems.
- ❑ We do not purchase and use COTS software

C. DEVELOPMENT PROCESS FOR BUILDING SYSTEMS USING COTS SOFTWARE

1. Please check the category that best describes the main development process you follow when building systems from COTS software

❑ Waterfall or linear strategy	❑ Evolutionary development
❑ V strategy	❑ Experimental development
❑ Throw away prototyping	❑ Spiral model
❑ Incremental development	❑ Other (*please specify*) ________________

2. Is your development process for building systems from COTS software iterative?

   ❑ No ❑ Yes

3. Please state any good practices, or lessons from past experience regarding the development process for building systems from COTS software

---

SECTION III. REQUIREMENTS ENGINEERING PHASE

A. TECHNIQUES FOR ACQUIRING AND SPECIFYING REQUIREMENTS

1. Indicate the techniques you use for acquiring and specifying requirements for systems from COTS software

	Never				Alwa
a. Use of brainstorming meetings and	1	2	3	4	5
b. Use of observation of current practices	1	2	3	4	5
c. Use of prototyping and user	1	2	3	4	5
d. Use of scenarios	1	2	3	4	5
e. Use of rich pictures and root definitions	1	2	3	4	5
f. Use of Soft System Methodology's conceptual models	1	2	3	4	5
g. Use of decision trees and tables	1	2	3	4	5
h. Use of entity-relationship modelling	1	2	3	4	5
i. Use of normalisation to structure data	1	2	3	4	5
j. Use of data dictionary	1	2	3	4	5
k. Use of entity life cycles	1	2	3	4	5
l. Use of data flow diagrams	1	2	3	4	5
m. Use of object and class diagrams	1	2	3	4	5

n. Use of structure diagrams	1	2	3	4	5
o. Use of matrices (e.g. Function/Event	1	2	3	4	5

p. Other (*please specify*): ____

2. Indicate the techniques you use for identifying COTS software

	Never				Alwa
a. Customer prior knowledge & past	1	2	3	4	5
b. Inventory of existing COTS software	1	2	3	4	5
c. Request for proposals (RFPs)	1	2	3	4	5
d. Market research	1	2	3	4	5
e. Software development fairs and shows	1	2	3	4	5
f. COTS software provider adverts and	1	2	3	4	5
g. Internet (Web) search	1	2	3	4	5

h. Other (*please specify*): ____

3. Please indicate the techniques you use to evaluate and select COTS components

	Never				Alwa
a. Study COTS software documentation	1	2	3	4	5
b. Attend demonstration by COTS software providers	1	2	3	4	5
c. Extensive experimentation with COTS	1	2	3	4	5
d. Customer prior knowledge & past	1	2	3	4	5
e. User community prior knowledge & past experience	1	2	3	4	5
f. Cards sorting and laddering	1	2	3	4	5
g. Feature analysis technique	1	2	3	4	5
h. Multi-criteria decision making	1	2	3	4	5
i. Outranking methods	1	2	3	4	5
j. Analytic Hierarchy Process (AHP)	1	2	3	4	5

k. Other (*please specify*): ____

4. Do you take into consideration the following issues when evaluating COTS software suitability (i.e. what evaluation criteria or framework do you use)?

	Never				Alwa
a. Compliant with essential customer	1	2	3	4	5
b. Software qualities (performance, resource requirements, reliability, etc)	1	2	3	4	5
c. Availability of good documentation and information on products	1	2	3	4	5
d. Maturity of the COTS software products	1	2	3	4	5
e. Maturity of the technology on which COTS software are based	1	2	3	4	5
f. Viability of the COTS software technology over the life of the system	1	2	3	4	5

g.	Price of the COTS software products	1	2	3	4	5
h.	Stability of COTS supplier	1	2	3	4	5
i.	Level of COTS supplier support	1	2	3	4	5
j.	Existing relationship with COTS supplier (partnership, credit agreement,	1	2	3	4	5
k.	Ease of migration to other products and technologies	1	2	3	4	5
l.	Political and economic factors	1	2	3	4	5
m.	Conformance to the appropriate standards and protocols (ODBC, HTTP,	1	2	3	4	5
n.	Ability to be tailored through the use of scripting & plug-ins	1	2	3	4	5
o.	Ease of integration through the use of the platform architecture (ActiveX, etc.)	1	2	3	4	5

p. Other (*please specify*): ____

B. TOOLS TO SUPPORT THE PROCESS OF ACQUIRING AND SPECIFYING REQUIREMENTS

1. Do you use any support tools (e.g. project management, drawing, CASE) during requirements acquisition and specification for COTS?

   ❑ No ❑ Yes

2. If yes specify tools used ____

3. Do you use any support tools (e.g. project management, drawing, CASE) during the COTS software identification and selection phase?

   ❑ No ❑ Yes

   If yes specify tools used ____

C. ORGANISATIONAL ISSUES

1. Do you involve customers in the following activities when building systems from COTS software?

		Never				Alwa
a.	Requirements acquisition	1	2	3	4	5
b.	Requirements specification	1	2	3	4	5
c.	COTS software identification	1	2	3	4	5
d.	COTS software evaluation	1	2	3	4	5
e.	COTS software selection	1	2	3	4	5
f.	Systems and architecture design	1	2	3	4	5

2. Do you take into consideration the following social and organisational issues when building system from COTS software?

		Never				Alwa
a.	Customer motivation and degree of	1	2	3	4	5

b. Customer education skills and	1	2	3	4	5
c. Differing goals, perception and communication by individuals working	1	2	3	4	5
d. Structure of organisation and internal	1	2	3	4	5
e. Changing organisational and business	1	2	3	4	5
f. Organisational resources and support	1	2	3	4	5
g. Organisational culture	1	2	3	4	5
h. Broader external environment that affects your company	1	2	3	4	5

3. In general how do you deal with the social and organisational issues?

D. GOOD PRACTICES WHEN ACQUIRING AND SPECIFYING REQUIREMENTS

1. Please indicate any good practices or lessons you have learnt from past experience when acquiring and specifying requirements for systems using COTS software (e.g. Ensure customer representation during COTS software evaluation)

SECTION IV. APPROACHES TO BUILDING SYSTEMS USING COTS SOFTWARE

A. PURCHASE AND USE APPROACH

In this approach you purchase a single complete working COTS software system that satisfies most of the user requirements and use it without adapting or extending it.

1. Please indicate the number of COTS software products you are using with this approach
   - ❑ 0 – 5
   - ❑ 5 – 10
   - ❑ 10 – 15
   - ❑ 15 – 20
   - ❑ 20 – 25
   - ❑ Above 25
2. Please check the box(s) that describe the main application of the COTS software products you are using with this approach. Select ALL applicable.
   - ❑ Office automation
   - ❑ Database systems
   - ❑ Accounting and finance
   - ❑ Email and messaging systems
   - ❑ GUI builders
   - ❑ Geographic Information Systems
   - ❑ Operating systems
   - ❑ Real time and embedded systems
   - ❑ Safety critical system
   - ❑ Business applications
   - ❑ Other applications (*please specify*): ______________________________
3. What problems have you experienced with the purchase and use approach?
   - ❑ COTS software not satisfying requirements
   - ❑ New releases of COTS software
   - ❑ Lack of support from COTS software providers
   - ❑ Other (*please specify*): ________________________
4. Please indicate any good practices or lessons you have learnt from past experience for the purchase and approach

B. PURCHASE AND ADAPT APPROACH

In this approach you purchase a single complete working COTS software system that satisfies most of the user requirements and then adapt or extend it for local needs.

1. Please indicate the number of COTS software products you have purchased and adapted (purchase and adapt approach)
   - ❑ 0 – 5
   - ❑ 5 – 10
   - ❑ 10 – 15
   - ❑ 15 – 20
   - ❑ 20 – 25
   - ❑ Above 25

2. Please check the box(s) that describe the main application of the COTS software products you have purchased and adapted. Select ALL applicable.
   - ❑ Office automation
   - ❑ Database systems
   - ❑ Accounting and finance
   - ❑ Email and messaging systems
   - ❑ GUI builders
   - ❑ Geographic Information Systems
   - ❑ Operating systems
   - ❑ Real time and embedded systems
   - ❑ Safety critical system
   - ❑ Business applications
   - ❑ Other applications (*please specify*): ______________________________

3. What methods (or techniques) are you using to modify and extend COTS software?
   - ❑ Application Programming Interfac (API)
   - ❑ Modify source code
   - ❑ Write plug-ins
   - ❑ Scripting (e.g. Javascript an VisualBasic)
   - ❑ Inheritance in object-oriente programming
   - ❑ Other (*please specify* __________________

4. What programming languages or development tools are you using to modify and extend your COTS software?
   - ❑ Java/Javascript
   - ❑ VisualBasic
   - ❑ Applescript
   - ❑ Perl
   - ❑ Ada
   - ❑ C/C++
   - ❑ Delphi
   - ❑ Other (*please specify*) __________________

5. What problems have you experienced with purchase and adapt approach?
   - ❑ Limited choice of supply of COT software
   - ❑ COTS software not satisfyin requirements
   - ❑ New releases of COTS software
   - ❑ High prices of COTS software
   - ❑ Lack of support from the COT software providers
   - ❑ Difficult to easily modify COT software
   - ❑ Other (*please specify*) __________________

6. Please indicate any good practices or lessons you have learnt from past experience for the purchase and adapt approach

C. COMPONENT INTEGRATION APPROACH

In this approach you purchase a number of COTS software components that satisfies some of the requirements of the system and build software system by integrating these components.

1. Please indicate the number of software systems you have built by integrating COTS software components (component integration approach)
   - ❑ 0 – 5
   - ❑ 5 – 10
   - ❑ 10 – 15
   - ❑ 15 – 20
   - ❑ 20 – 25
   - ❑ Above 25

2. What component types do you normally integrate to build your systems?
   - ❑ Procedural libraries
   - ❑ Legacy applications
   - ❑ Off-The-Shelf applications
   - ❑ Tools e.g. GUI builder
   - ❑ System services like database or operating system
   - ❑ Frameworks
   - ❑ OLE objects from Microsoft.
   - ❑ Other (*please specify*) ________________

3. Please check the box(s) that describe the main application of the systems you have built by integrating COTS software components. Select ALL applicable.
   - ❑ Office automation
   - ❑ Database systems
   - ❑ Accounting and finance
   - ❑ Email and messaging systems
   - ❑ GUI builders
   - ❑ Geographic Information Systems
   - ❑ Operating systems
   - ❑ Real time and embedded systems
   - ❑ Safety critical system
   - ❑ Business applications
   - ❑ Other applications (*please specify*): ______________________________

4. What technologies and standards do you normally use to support component integration?
   - ❑ Component Object Model (COM) from Microsoft
   - ❑ DCOM
   - ❑ Object Linking and Embedding (OLE)
   - ❑ Dynamic Data Exchange (DDE)
   - ❑ ActiveX
   - ❑ Common Object Request Broker Architecture (CORBA)
   - ❑ OpenDoc
   - ❑ Open Scripting Architecture (OSA)
   - ❑ RMI
   - ❑ Other (*please specify*):__________________

5. What type of architectural styles are you using for developing systems from COTS software components?
   - ❑ Procedure calls such as application with API or database with SQL interface
   - ❑ Desktop supported facilities such as drag and drop, clipboards, cut and paste
   - ❑ Message bus with components having separate data stores co-ordinated through message announcements
   - ❑ Data sharing such as shared database and file
   - ❑ Object request broker (ORB) mediated

- ❑ Other (*please specify*) ____________________

6. What problems have you experienced with integrating COTS software components?
   - ❑ Lack of stability and support from provider
   - ❑ New releases of COTS products
   - ❑ Lack of information about COTS products
   - ❑ Difficult to easily integrate components
   - ❑ Conflicting standards for component integration
   - ❑ Other (*please specify*) ____________________

7. Please indicate any good practices or lessons you have learnt from past experience for this approach

8. Do you normally use support tools (e.g. CASE) in COTS software integration?
   ❑ No ❑ Yes
9. If yes, *please specify* tools used: ____

Thank you for completing the questionnaire. Your assistance is much appreciated. Any information disclosed will remain STRICTLY CONFIDENTIAL. The results of the survey will be in the form of aggregated data (synthesis) and no answers will be associated with individual respondents. A report summarising the results of this survey will be available to all respondents. If you would like a copy please check the yes box.

❑ Yes ❑ No

You may return the questionnaire to the address below. If you have any queries about the questionnaire or the nature of this research please do not hesitate to contact:

Douglas Kunda
Department of Computer Science, University of York

12 Appendix 4: Interview protocol and questions for study 2

Interview protocol to identify factors that support COTS components evaluation for COTS-Based Systems

The interview protocol was divided into five sections. Section 1 provided the background information, the theoretical framework and the objective of the field study (presented in thesis as chapter 2). Section 2 described the key features of the field study method. It presented the field study research design, methodology, data collection and study database (presented in thesis as part of chapter 5). Section 3 outlined the field procedures (credentials and access to the field study sites, general sources of information and procedural reminders). Section 4 presented the interview protocol itself and the questions. Section 5 discusses the field study analysis plan and study report format.

In this appendix only a summary of the questions used during the interviews have been reproduced. The other materials are available in the parts of this thesis.

INTERVIEW QUESTIONS

Sources of Data:

Obtain internal guidelines or standards for COTS evaluation (or CBS)
Obtain reports of previous evaluation (if available)
Obtain documented examples of evaluation criteria used by the company

Contextual Questions

Introduce myself and COTS, my name is Douglas Kunda a research student with interest in Commercial-Off-The Shelf (COTS). The term "COTS" is meant to refer to things that one can buy, ready-made, from some manufacturer's virtual store shelf (e.g., through a catalogue or from a price list). For example Microsoft Access. CBS focuses on building large software systems by integrating previously-existing software components. COTS software selection also known as component qualification is a process of determining "fitness for use" of previously-developed components that are being applied in a new system context. *The outline of this interview*, we will initially discuss the evaluation criteria used, then the search process for COTS products, and lastly the evaluation or assessment process. The interview will last approximately one hour.
(Q1) Before we proceed into the detailed discussion let me begin by asking, what is your responsibility in your organisation?
(Q2) Have you participated in selecting COTS software components that where later adapted or integrated into systems? Approximately how many times? Why?
An example is to adapt Microsoft Access database for your local needs or integrating acrobat reader with a browser through plug and play.

Criteria Definition

Technology factors

(Q3) What variables or factors do you consider important that influence the selection of particular COTS underlying technology such as CORBA or DCOM? How do you evaluate these qualities?
(Q4) Do you normally separate the evaluation of underlying technology from product evaluation? Does this make the selection easier or harder? Why?
Examples of techniques for evaluating these qualities are discussed later but they include documentation, experimentation, vendor demonstration, and user group advice.
Examples of technology factors include quality characteristics (performance issues such as dependability and resource utilisation, availability of documentation, security issues), and integrability (support of standards and protocols, adaptability, architectural and interface issues, support for plug and play, support for debug and testing).

Product quality factors

(Q5) ISO/IEC 9126 defines software quality characteristics as a set of attributes of software products by which its quality is described and evaluated. What quality characteristics or factors do you consider important that influence the selection of particular COTS products or technology? How do you evaluate these qualities?
Examples of quality characteristics are performance (dependability, efficiency, usability, reliability), maintainability and adaptability (interoperability, portability, scalability, reusability, adaptability, replaceability).

Compliance issues
(Q6) Compliance issues or functionality factors assess the product or technology conformance to requirements. What functionality variables or factors do you consider important that influences the selection of particular COTS products or technology? How do you evaluate these qualities?
Examples of functionality characteristics include suitability, accuracy, standards, security, safety requirements, and disaster characteristics.

Business issues
(Q7) What business or financial issues do you consider important that influence the selection of particular COTS products or technology? How do you evaluate these qualities?
Examples of business issues include cost of product/technology, licensing arrangements, additional cost of adapting and integrating the products, training and support costs, cost of maintenance or replacement with upgrades.

Customer capability
(Q8) Sometimes the customer capability may influence the selection of a product or technology. What customer capability variables or factors do you consider important that influences the selection of particular COTS products or technology? How do you evaluate these qualities?
Examples of customer capability include customer experience with product/technology, customer expectations, internal organisational politics, and customer/organisation policies or preferences.

Marketplace variables
(Q9) Commercial market place such as trends may influence the choice of products or technologies. What commercial marketplace variables or factors do you consider important that influences the selection of particular COTS products or technology? How do you evaluate these qualities?
Examples of marketplace variables include product/technology reputation (maturity, stability), product/technology restrictions, market trends (viability).

Vendor capability variables
(Q10) Sometimes vendor performance and capabilities may be included in the evaluation criteria. What vendor variables or factors do you consider important that influence the selection of particular COTS products or technology? How do you evaluate these qualities?
Examples of vendor variables include vendor reputation, certification, stability, available training and support.

Search for alternatives

Techniques and tools

(Q11) What important techniques do you consider to make the search for candidate products or technology easier and quicker? Why?

Examples of techniques and tools for searching for candidate products are market survey, vendor promotions (market fairs and shows), Internet search, user group and mailing lists, and request for proposals (RFP).

COTS availability

(Q12) How do you deal with the situation where there are too many products that satisfy your search criteria or a situation where there are fewer available products that fully satisfy your search criteria?

Assessment (Evaluation)

Evaluation strategy

(Q13) In COTS evaluation there are normally three strategies that is followed namely; progressive, keystone identification and puzzle, what COTS evaluation strategy do you normally adopt? Why?

In progressive filtering you start with a candidate set of components, progressively more discriminating evaluation mechanisms are applied in order to eliminate less "fit" components. Keystone selection strategy normally constrain the evaluator's freedom in selecting other products, since keystone characteristics (e.g., vendor, type of technology, API) will force a focus on specific characteristics of remaining components. The puzzle assembly model begins with the premise that a valid COTS solution will require fitting the various components of the system together as a puzzle.

Techniques and tools

(Q14) What data collection techniques and tools do you consider important that makes the selection of particular COTS products or technology easier and quicker? Why?

(Q15) What data analysis techniques and tools do you consider important that makes the selection of particular COTS products or technology easier and quicker? Why?

Examples of data collection techniques include study documentation, experimentation user group advice and attend demonstrations. Other techniques and tools include templates, checklists, questionnaires, card sorting, algorithms for benchmarks testing. Examples of data analysis techniques include Analytic Hierarchy Process (AHP) weighted sum method (WSM) and qualitative methods

Customer participation

(Q16) Does the participation of customers participate improve or impair COTS evaluations? Why? If it improves the evaluation process what type of participation (consultative, representative or consensus)?

Consultative participation is the lowest level of participation in which the users departments are simply consulted. Representative participation is a higher level at which the user department is represented in the evaluation team. Consensus attempts to involve all user departments throughout the process, this is user driven.

(Q17) In what phases of COTS selection process (evaluation, criteria definition, search for products) does customers participate improve or impair the selection process? Why?

General and conclusion

(Q18) What standards or guidelines do you follow when evaluating COTS products and technology examples include your own internal company guidelines, previous evaluation cases, IEEE std 1209-1992, ISO/IEC 9126:1991, SEI Technology Delta or simply your past experience?
(Q19) Does the use of evaluation criteria make it easier or harder to assess COTS products and underlying technology? Why, please give examples?
(Q20) Does the evaluation criteria influence or not the identification of candidate products and technology? Why, please give examples?
(Q21) Does the identification of candidate products and technology make it easier or harder to define the criteria? Why, please give examples?
(Q22) Does the identification of candidate products and technology improve or impair the assessment of COTS products and technology? Why, please give examples?
(Q23) Does the assessment of COTS products and technology influence or not the criteria definition? If yes in what ways, please give examples?
(Q24) Does the assessment of COTS products and technology influence or not the identification of candidate COTS products and technology? If yes in what ways, please give examples?
(Q25) What best practices would like to recommend that can assist in improving the COTS evaluation process and CBSD in general?

Thank you very much for your assistance. I want to assure you once again regarding the confidentiality of the data you have submitted that it will only be used in a thesis and academic publication in aggregated manner. A draft case study report will be available for you to make comments.

The documents to be collected include guidelines to CBS development, guidelines to COTS software evaluation process, evaluation criteria used, vendor documentation evaluated, reports or case study reports regarding past COTS evaluation, URLs sources for COTS components and other sources for COTS components.

Sample letters of access procedures for study 2

Dear Sir/Madam,

SECOND STUDY: COTS SOFTWARE COMPONENT EVALUATION AND SELECTION

I refer to your mail dated 19th August 1998 in which you indicated willingness to assist me with the second study. I would like to thank you for your time and co-operation in this research effort.

As I indicated the overall goal of this study is to identify best practices from the industry in the UK regarding evaluation and selection of Commercial-Off-The Shelf (COTS) software components for CBS. The following are the immediate objective of the second study:

- Identify problems (and solutions) experienced by companies in evaluating COTS components for CBS and define criteria for judging COTS evaluation success;
- Explore strategies for COTS software evaluation and analyse how they foster evaluation success; elicit techniques and tools for evaluating COTS components and consider in what ways they promote evaluation success;
- Review the COTS components evaluation criteria used by companies (whether it includes the social-economic factors) and investigate how the evaluation criteria definition can be improved; and
- Investigate in what ways customer participation can contribute to COTS components evaluation success.

I would like to make phone appointment for Friday 23rd October 1998 in the morning or any other day and time convenient to you to discuss the details of the study. During this phone discussion I brief you about the study and we will agree on dates when I can visit your institution to conduct the interview. You could confirm this phone appointment on the address above or by email to douglas@cs.york.ac.uk. Thanking you in anticipation of your future co-operation.

Let me assure you that your organisation's response will be kept strictly confidential. No data will be associated with individual respondents or organisations and the results will only be analysed and reported in aggregate form.

Yours faithfully,

Douglas Kunda
Research Student

13 Appendix 5: STACE framework[*]

The STACE framework has been developed to facilitate a simple, quick and easy to use social-technical approach to COTS selection process. The STACE framework has been developed through literature survey and studies in the UK. STACE is based on a number of important principles and these are:

- Support for a *systematic approach* to COTS evaluation and selection (Kontio, 1996). Most organisations select their COTS components in an ad-hoc manner. There is need to reuse lessons learnt from previous evaluation cases by maintaining a database of evaluation results.
- Support for *evaluation of both COTS products and the underlying technology*. Most COTS evaluation frameworks emphasise either on COTS products evaluation or technology evaluation. This method proposes keystone evaluation strategy (Obarndorf, et al, 1997) in which the underlying technology is selected before selecting the COTS products.
- Use of *social-technical techniques* to improve the COTS software selection process. This has been greatly influenced by the social-technical school and work by (Mumford, 1995). The STACE recommends the use of a social-technical evaluation criteria and customer participation in the COTS selection process. User participation is regarded as an effective strategy for improving software design outcomes and as a means of incorporating human and organisational aspects such as the design of jobs, work processes and usability (Bravo, 1993)(Axtell, et al, 1997).
- Use of *multi-criteria decision-making techniques* to consolidate evaluation attribute data. The STACE proposes the use of Analytic Hierarchy Process (AHP) which was developed by Saaty (1990) and successfully used in software selection (Zviran, 1993)(Kontio, 1996).

The STACE framework (see figure 1) comprise four interrelated processes: 1) requirements definition; 2) social-technical criteria definition; 3) alternatives identification; and 4) evaluation or assessment.

[*] The references for this appendix are incorporated in the main reference section of this thesis.

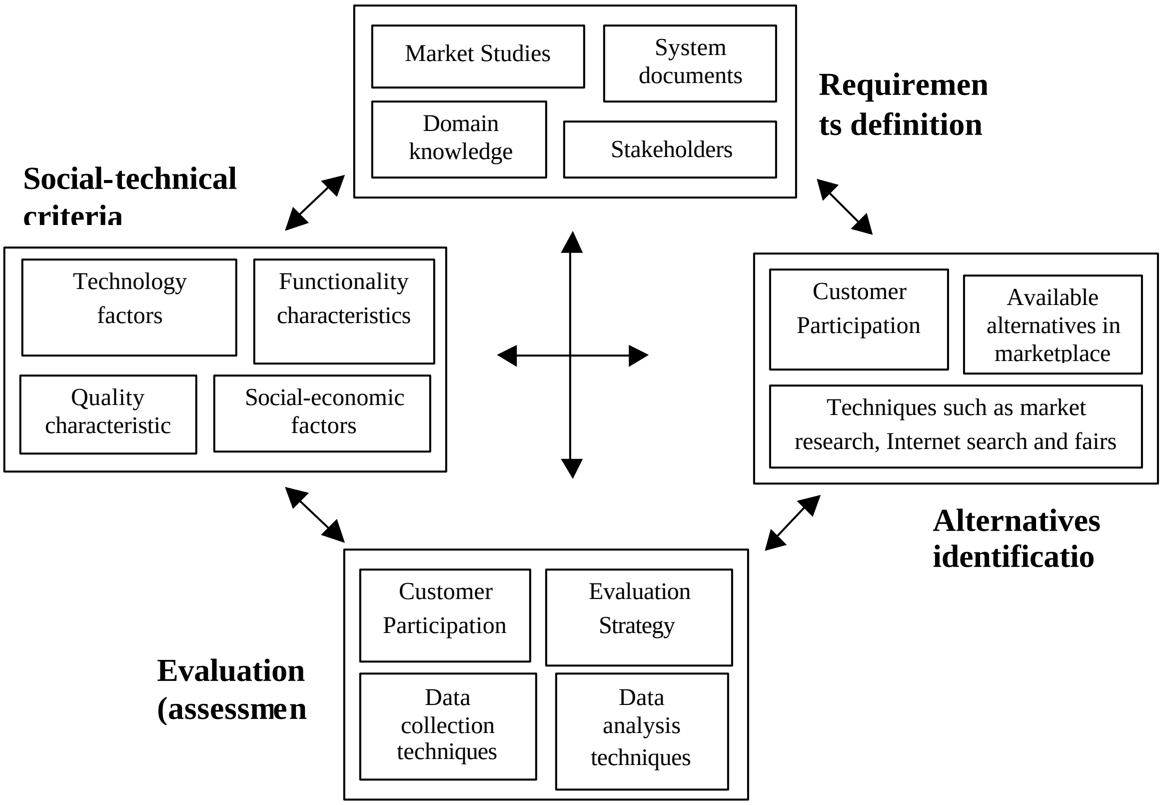


Figure 2: STACE Framework

Requirements definition

In the requirements elicitation process, the high-level customer and systems requirements are discovered through consultation with stakeholders, from system documents, domain knowledge and market studies (Sommerville, 1995). The use of COTS products introduces new problems for requirements engineers, for example, deciding when to acquire new customer requirements and when to reduce the number of candidate products (Maiden and Ncube, 1998). Dean and Vidger (1997) propose acquiring a small number of high-level requirements prior to an iterative and concurrent product evaluation and selection. Finkelstein et al (1996) points out that the acquisition process must focus on requirements that discriminate most between COTS products. Therefore, the STACE framework recommends eliciting high level requirements because stakeholders often have prior knowledge of candidate products that they prefer. This save on resources by not spending too much time describing in detail the requirements of a system that exists in the marketplace (Oberndorf, 1997).

In order to realise the benefits of COTS software, a procurement process must be in place that defines requirements according to what is available in the marketplace, and that is flexible enough to accept COTS solutions when they are proposed (Vigder, et al, 1996). The problem with a procurement process that identifies strict requirements is that either it will exclude the use of COTS components, or will require large modifications to COTS packages in order to satisfy the requirements. Therefore, STACE framework recommends that the high-level requirements be partitioned according to the types of packages expected to be available in the relevant problem domains. The requirements must be adjusted to maximize package use and architecture created that promotes the use of acquired packages. This is a paradigm shift from custom development where the system requirements drive capabilities. However, to avoid a bias risk, STACE recommends that evaluators must be careful not to redefine requirements so specific that only one particular product is suitable.

Many researchers for example Maiden and Ncube (1998) and industry (SEL, 1996) recommend the use of scenarios and use-cases as a way of specifying the behaviour of the systems and suggests that this fits well with the CBS approach. The form of documenting use-cases can range from simple text descriptions to elaborate descriptions of processing scenarios that include prototype screens. The advantages of use-cases include ease of understanding and communication with users, user-centred development, simplifying the mapping between requirements and the proposed COTS and reusable packages (Jacobson, 1995). The STACE method recommends the identification of use-cases that define operational scenarios for the system and mapping them to the high-level requirements.

The traditional requirements engineering methods emphasise the technical issues whilst neglecting the equally important social issues (Jirokta and Goguen, 1994). The STACE method recommends the use of the social-technical approach to systems development. A social-technical development method is a method to develop a system that consists of the human subsystem and a technical subsystem in an integrated way (Wieringa, 1996). Taylor and Felten (1993) points out that sociotechnical systems require the participation of system members and that involvement provides the content of, and the reasons for, empowerment. In STACE method, customer participation is achieved through JAD sessions and review meeting with top management. JAD centres around one structured workshop session in which everyone gets together in a room and "talks it out" and everyone hears what the rest of the group has to say. More detailed discussion of JAD sessions is provided in Wood and Silver (1995).

The outcome of the requirements definition phase are a high-level architecture (including candidate packages and use-cases) and refined requirements are the products of this phase.

Social-technical criteria definition

The evaluation criteria are parameters against which the COTS product is evaluated and upon which selection decisions are made (IEEE, 1993). In the evaluation criteria definition process, the high-level requirements from the requirements definition phase are decomposed into a hierarchical criteria set and each branch in this hierarchy ends in an evaluation attribute (Kontio, 1996). The STACE framework uses decomposition approach that is based on social technical analysis and the AHP criteria decomposition method. The requirements definition is influenced by what products are available in the marketplace, therefore the criteria will also be influenced by what is available in the market. Another possible source of information in defining the criteria is the experience from past evaluation cases.

The evaluation must be based on essential requirements, not optional or the "nice to have" requirements (Kontio, 1996)(Beus-Dubic and Wellings, 1998). Therefore, the social-technical criteria must be tailored based upon customer needs and priorities. The framework recommends using some existing standard checklist or template to define the criteria. In the STACE framework, the social-technical criteria include: 1) technology factors, 2) functionality characteristics, 3) product quality characteristics, and 4) social-economic factors.

Technology factors - A technology is a specification or framework that provides for integrating components. The COTS component underlying technology is the basis for the component's interoperability, portability, reusability, maintainability and adaptability. The COTS underlying technology is selected from high-level customer requirements. For example, the customers could specify that they prefer a system that is based on CORBA technology. However, an understanding and evaluation of the underlying assumptions about the technology must be elicited and analysed before commitment to a particular technology. In literature (Haines, et al, 1997)(Szyperski, 1998), a number of issues have been proposed for consideration and evaluation when selecting a particular technology. The STACE framework recommends:

- *Functionality* - functional requirements that the technology should support for example support for distributed objects, platform support, real time processing.
- *Performance* - the quality measures that address how well a technology function such as dependability, efficiency, resource utilisation, and usability. For example the way the technology handles memory management issues can be assessed.
- *Framework and architecture style* - the type of infrastructure that provides the binding from disparate components such as object request broker mediated, message bus, database and blackboard.
- *Interface standard* - interfaces are the means by which components connect and interact. The degree to which a software component meets certain standards can greatly influence the interoperability and portability of a system.
- *Security issues* - the capability of a technology to manage, protect and distribute sensitive information.

Functionality characteristics - ISO/IEC 9126:1991(E) defines functionality as a set of attributes that bear on the existence of a set of functions and their specified properties. The functions are those that satisfy stated or implied needs. Kontio (1995) points out that the functionality criteria are derived from requirement of design specification and expressed in form of requirements. In the STACE framework, it is recommended that the functional requirements be represented in the form of scenarios or use-cases and includes essential customer requirements and customer standards. The functionality characteristics help in the initial selection of candidate COTS software that will be evaluated. For example, Morisio and Tsoukias (1997) in selecting a CASE tool to support the production of software in CIM environment used the following functionality attributes:

- *Editing*, the possibility of editing the model
- *Executing*, the possibility of executing the model
- *Data analysis*, the possibility of analysing the data produced by the executing model
- *Debugging*, the possibility of debugging the model from inside the product
- *Simulation*, the possibility of simulating the model and this was decomposed into data collection, statistical libraries, data structure libraries, and graphical analysis
- *Software generation*, the possibility of generating software from the model and this was decomposed into field interfaces, database interfaces and graphical interfaces.

Product quality characteristics - ISO/IEC 9126:1991(E) defines software quality characteristics as a set of attributes of software products by which its quality is described and evaluated. COTS component quality characteristics are behavioural properties that the product must have and should match the customer's non-functional requirements (Kontio, 1996). The product quality characteristics do not necessary change from application to application. However, the STACE framework recommends a review and adaptation of these quality attributes in accordance to customer requirements and priorities. Examples of the quality characteristics include (IEEE, 1993) (ISO/IEC 9126, 1991):

- Efficiency, the degree to which a system or component performs its designated functions with minimum consumption of resources (CPU, Memory, I/O, Peripherals, Networks).
- Interoperability, attributes of the software that bear on its ability to exchange information and to use the information that has been exchanged.
- Maintainability, the ease with which a software system or component can be modified to correct faults, improve performance, or other attributes, or adapt to a changed environment.
- Portability, the ease with which a system or component can be transferred from one hardware or software environment to another.
- Reliability, the ability of a system or component to perform its required functions under stated conditions for a specified period of time.
- Usability, the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component.

Socio-economic factors - these are non-technical factors that should be included in the evaluation and selection of COTS components such as costs, business issues, vendor performance and reliability. Costs include direct costs, such as the price of the COTS software products, and indirect costs, such as the cost of adapting to local needs as well as training costs. Organisational issues include people and process problems that must be overcome before successfully implementing the COTS based system, such as management support and internal organisational politics, staff skills and attitudes. Vendor performance and reliability includes vendor infrastructure and stability, period of vendor business, vendor reputation, references, customer base and track record. These are most frequently overlooked factors that bring in the social dimension in the evaluation criteria. STACE framework recommends a number of social-economic issues must be considered when selecting a product or technology. They include:

- *Business issues* - the financial case for buying a particular product or technology (Powell, et al, 1997). Business issues include cost of product/technology, licensing arrangements, additional cost of adapting and integrating the products, training and support costs, cost of maintenance or replacement with upgrades.
- *Customer capability* - examples of customer capability include customer experience with product/technology, customer expectations, internal organisational politics, and customer/organisation policies or preferences.
- *Marketplace variables* examples of marketplace variables include product or technology reputation, maturity and stability of a product (Klopping and Bolgiano, 1990), product or technology restrictions, market trends and viability of products over long period (Rowley, 1993).

- *Vendor capability variables* - the performance and capability of the vendor (Hokey, 1992) and examples include vendor profile, reputation, certification, stability, available training and support.

Identifying candidate products (alternatives identification)

The objective of this process is to identify COTS components that meet the high level requirements so that they can be considered for a more rigorous evaluation (Tran, et al, 1997). In STACE framework, this phase begins with identifying the domains relevant to the problem and understanding the types of packages available in those domains. This followed by the development of a high-level architecture that maps "non-solution-specific" package types to high-level requirements. A high-level architecture in this context is simply an arrangement of the available types of domain packages into a structure deemed appropriate for a solution, without a judgment yet as to the quality of that solution (SEL, 1996). The use-case transforms are then mapped to the proposed COTS and reusable packages. The next step is to obtain information about the COTS products. This information may consist of evaluations done by independent evaluators, reports from vendors, vendor demonstration of the product capabilities and information obtained directly from actual users (Rowley, 1993).

Kontio (1996) argues that it is important to screen and reduce the candidate products to reasonable and manageable number for later detailed evaluation by conducting a preliminary "paper" evaluation. Maiden and Ncube (1998) recommend the use of templates to reduce the number of candidate products. Each template defines the product information and customer requirements to acquire and the techniques for acquiring this information and making decisions about it. In addition to "paper" evaluation and template-based method, the STACE framework recommends the use of elimination by aspect method. Elimination by aspect examines one attribute at a time, making comparisons among alternatives (Yoon, 1995). It eliminates alternatives that do not satisfy some standard and it continues until all alternatives except three or four have been eliminated. The elimination is based on the most important attributes and also those attributes that would eliminate the most alternatives.

There are a number of factors and processes that impact on successful identification of candidate products including the definition of requirements, availability of products in marketplace, adopted techniques and tools. The STACE method recommends a number of techniques and tools for identifying candidate COTS products. These include:

- Networking, mailing list and user community (Kontio, 1995). In this method individuals of the evaluation team will draw from experience and networking capability in order to identify available COTS products in the marketplace. The mailing list and product user community can be used to elicit more information on available products and their capabilities.
- Internet search (SEL, 1996). Internet search using search engines is one of the most effective methods for identifying available COTS products in the marketplace. In some cases, vendors make available demonstration copies of the COTS products on the Internet, which can be downloaded and evaluated. STACE framework have a number of suggestion on the effective searching of the Web, for example using a metasearch engine (or simply repeating a search at multiple engines) for harder to find information.

- Market surveys (Rowley, 1993)(Maiden and Ncube, 1998). A market survey can be made using questionnaires in order to identify available COTS products in the market. These methods pre-suppose the availability of vendor mailing list or address book.
- Invitation to tender (ITT) or request for proposals (RFP)(Finkelstein, et al, 1996). Most public institutions use the tender procedures as a method to identify the available products and their capabilities. An advert is placed on the Internet or newspapers and organisations are requested to bid for the tender.
- Vendor promotions and publications (Tran, et al, 1997). Search can be made through vendor publications and catalogues in order to identify available COTS products. Members of evaluation teams can also arrange to attend computer fairs and shows in order to identify what COTS products are available in a particular domain.

The outcome of this phase is candidate COTS software products. STACE recommends obtaining demonstration copies of candidate products and their supporting documentation.

Evaluation (assessment)
The evaluation involves contacting vendor technical support for evaluation information, review of vendor documentation and product testing for quality and functionality (Vigder, et al, 1996)(Kontio, 1996). It includes evaluating COTS performance, interfaces and ease of integration, comparing short-term and long-term licensing costs against integration costs. The reasons for selecting each component and the reasons for rejecting others are recorded (Maiden and Ncube, 1998). The outcome of the evaluation process is the recommended COTS product or products. A number of evaluation strategies have recommended in literature such as progressive filtering, keystone identification and puzzle assembly (Oberndorf, et al, 1997). STACE recommends the keystone identification strategy with the technology as the keystone issue. In keystone selection strategy, a keystone characteristic such as vendor or type of technology is selected first before selecting the COTS products (Walters, 1995). The separation of COTS underlying technology from COTS products during evaluation allows fair comparisons between products. The other advantage of separating products from technology is that useful literature is available on technology comparisons, since technology change is not as fast as product change.

STACE framework recommends separating the data collection and data analysis of the evaluation. Kontio (1996) argue that the advantage of separating the data collection from analysis is to allow the use of appropriate decision making techniques in data analysis stage. There are a number of data collection techniques such as examining the products and vendor supplied documentation, viewing demonstration and interviewing demonstrators, executing test cases and applying the products in pilot projects (Tran, et al, 1997). STACE proposes selecting appropriate techniques depending on resources and experience. In addition data collection may include interviewing actual users of the products, and examining sample outputs from projects that have used the products. Other data collection techniques include vendor analysis (Hokey, 1992), auditing the development process used to develop the software including tests carried out (Haines, et al, 1997).

STACE recommends the use of the Analytic Hierarchy Process (AHP) to consolidate evaluation data in order to select the "best" components among alternatives. There are tools available to support the AHP techniques. AHP was developed by Saaty (1990) for multiple criteria decision making and has been successfully used in software selection (Hokey, 1992)(Maiden and Ncube, 1998). The AHP technique is based on pair-wise comparison between alternatives. The result of this pair-wise comparison is converted to a normalised ranking by calculating the eigenvector from the comparison matrix's largest eigenvalue. The advantages of the AHP technique include a systematic approach for consolidating information about alternatives using multiple-criteria; an objective weighing technique for setting the weighing scale for qualitative and quantitative data, and it allows for consistency checking (Kontio, 1996)(Zviran, 1993).

14 Appendix 6: STACE Method workbook (second version)[2]

The references for this appendix are incorporated in the main reference section of this thesis

1. INTRODUCTION

COTS software component selection is a process of determining "fitness for use" of previously-developed components that are being applied in a new system context (Haines et al, 1997). Component selection is also a process for selecting components when a marketplace of competing products exists. Selection of a component can also extend to include qualification of the development process used to create and maintain it (for example, ensure that algorithms have been validated, and that rigorous code inspection has taken place).

The Social-Technical Approach to COTS Evaluation (STACE) framework was developed to facilitate a simple, quick and inexpensive social-technical approach to COTS selection process. The STACE framework (see Figure 1) comprises of four processes that are dependent on each other: 1) requirements elicitation; 2) social-technical criteria definition; 3) alternatives identification; and 4) evaluation or assessment. This indicates that each process can provide feedback on the other processes and the selection will involve multiple iterations through these processes.

[2] Note that the additional text to the first version of the workbook is underlined this second version

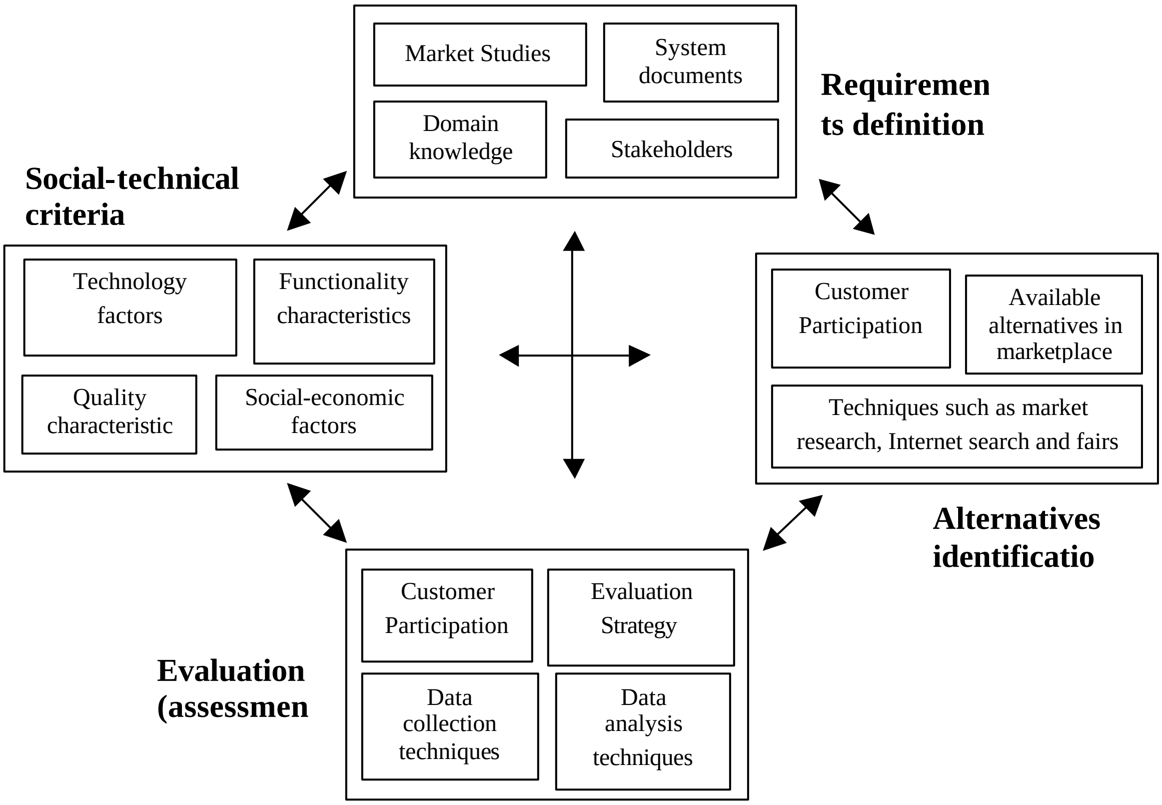


Figure 1: Shows the STACE Framework

In requirements elicitation process the high-level customer and systems requirements are discovered through consultation with stakeholders, from system documents, domain knowledge and market studies (Sommerville, 1996). In social-technical criteria definition process essentially the high-level requirements from the requirements elicitation are decomposed into a hierarchical criteria set and each branch in this hierarchy ends in an evaluation attribute (Kontio, 1996). Alternative identification includes searching and screening for COTS products/ technology that will be assessed in the evaluation stage. This process is driven by guidelines and criteria defined in the criteria definition process. Evaluation process involves ranking of identified COTS alternatives against the social-technical evaluation criteria by examining capabilities, reading documentation and experimentation. STACE framework is also presented in flow diagram (figure 2).

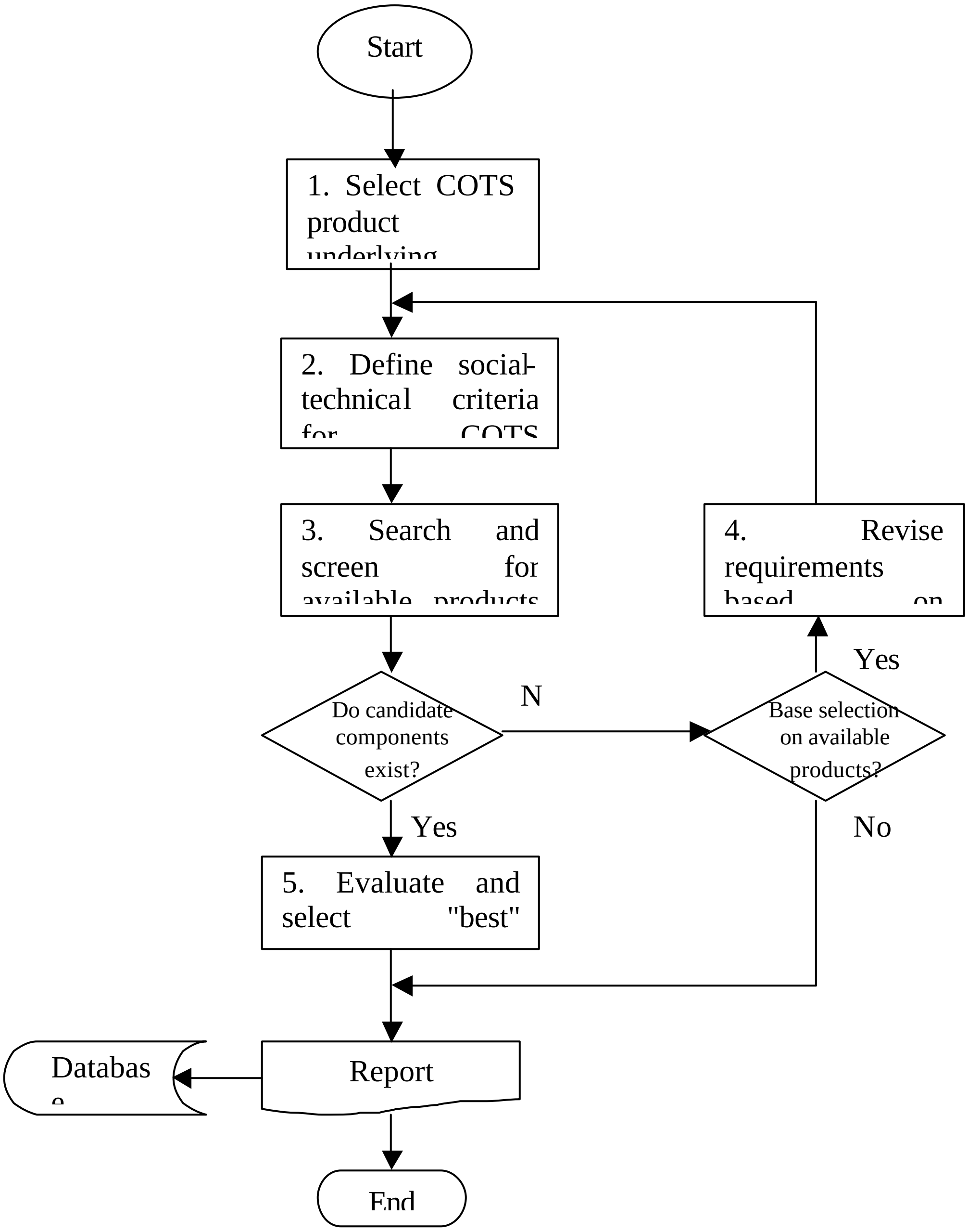


Figure 2: Evaluation process

2. PROBLEM/ REQUIREMENTS DEFINITION

Requirements definition highlights	
Input:	Customer requirements Stakeholders
Output:	Evaluation team set up Requirements Definition Document (RDD)
Checkpoint:	Requirements Definition Review (RDR)
Steps:	1. Constitute evaluation team

2. Define problem and outcomes (requirements)
3. Develop high-level system architecture

2.1 Overview

The purpose of the requirements definition phase is to produce a clear, complete, consistent, and testable specification of the social-technical requirements for the software product. Requirements definition initiates the COTS software evaluation and selection process. During this phase, the evaluation team uses an iterative process to expand a broad statement of the system requirements into a complete and detailed specification of each function that the software must perform and each criterion that it must meet. The finished requirements and specifications, combined with the system and operations concept, describe the software product in sufficient detail so that even if independent software consultants were used they can select the required package correctly. The first step is partitioning the high-level requirements according to the types of packages expected to be available in the relevant problem domains. To avoid a bias risk, they must be careful not to define requirements so specifically that only one particular product is suitable.

The team sets out deliberately to create an architecture that will promote the use of acquired packages. This architecture will incorporate patterns based on the experience of team members with other systems that use packages. Reuse of appropriate high-level architectures is critical at this juncture. The team should expect to review and refine several strawman architectures to arrive at a reasonable candidate architecture that can be mapped to available packages running on suitable platforms. The team identifies use-cases that define the operational scenarios for the system and maps them to the requirements. (Use-cases are simply a way of putting structure around an operational scenario and mapping it more directly to software components and test cases.) At the end of the phase, a requirements definition review is held with management to obtain approval. The requirements definition document is the major output of this phase.

2.2 Key Activities

Constitute the evaluation team. Identify the decision-maker and stakeholders from the organisation. Agree with the decision-maker regarding the composition of the evaluation team and the resources required for the evaluation work. The decision-maker is the person (or persons) who has both authority and the need to select a COTS product.

> Identify the decision-maker and stakeholders for example of COTS software selection project that you undertook or about to undertake:
>
> **Stake holder Name** **Stakeholder position**

Define the problem and outcomes of the evaluation. Define the problem to be solved by the process and the possible outcomes. The problem can be elicited from the requirements specification document. In the event that the requirements specification document is not available, the team should elicit the requirements from system documents, domain knowledge, and stakeholders using Joint Application Development (JAD) workshops. An initial assessment of the decision risk and benefit analysis can be made at this time. The outcome of this process would be high level requirements.

> Define the problem and outcome of the evaluation
>
> *Activity 2.2*

Identify the use cases (or scenarios) from the high level requirements. Identify use-cases (or scenarios from the high level requirements). A use-case is a way of specifying the behaviour of the systems and it fits in naturally with the COTS

> Draw the high-level use-case for the problem and outcome identified in activity 2.2
>
> *Activity 2.3*

approach.

Develop high-level system architecture and requirements definition document. Prepare a high-level diagram that shows the major components of the system and how

> Draw a high-level system architecture of your proposed system (full page)

the COTS software will be linked to satisfy the system requirements. Finally produce the requirements definition document.

2.3 Methods and Tools

Joint Application Development (JAD) workshop

This is a workshop or structured meeting in which users and technical developers work together to elicit requirements. The objective of JAD is to speed up the software development process and focuses upon identifying the important users and involving them via workshops at early stages of development. Participation may lead to increased user acceptance by developing realistic expectations about the systems capabilities, providing an arena for bargaining and conflict resolution about the selected product and leading to system ownership by users. Other benefits of the workshop include:

- Elapsed time is shorter
- Issues raised and dealt with
- Ambiguities less likely because issues are clarified in the meeting
- Confidence levels not compromised
- Agreed and documented output
- Team building and empowerment of individuals

Typical characteristics of JAD workshop include: an intensive meeting of business users and information systems people; a defined length of meeting; a structured meeting room and a facilitator and scribe. Full description of the JAD process is documented in Wood and Silver (1995).

Use-Cases identification and UML CASE tool

A use-case is a way of specifying the behavior of the system as a dialog between the user and the system. It is a technique, which has evolved over the last few years in the software industry, that fits in naturally with the COTS approach. Structuring project activities around use-cases helps center the team on the user's perspective of the packages to be included in the system and provides a natural framework for package evaluation and later incremental development and system testing. The form of documenting use-cases can range from simple text descriptions to elaborate descriptions of processing scenarios that include prototype screens. A use-case model is a graph with two types of nodes, actor (consultant) nodes and use-case nodes (word processing) in figure 3. Use-cases play an important role in driving the whole development work and simplifying the mapping between requirements and the proposed COTS and reusable packages.

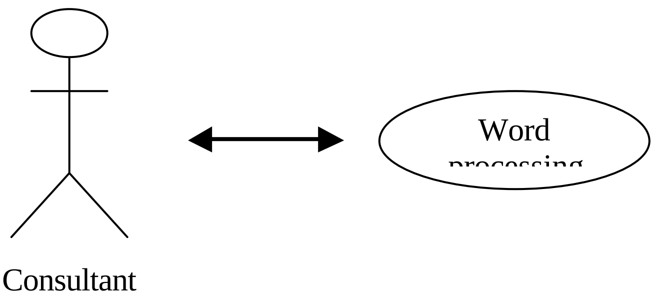


Figure 3. Use case notation

The team identifies the system products and inputs for each use-case, then determines the transforms for generating the expected outputs from the given inputs. The transforms map the user's actions on the system to the system's response to those actions. Use-cases should focus on what the user needs to achieve not how they are currently doing it. In other words, the goal is not to simply automate an existing process, but to improve the entire activity. A CASE tool supporting UML notation such as Rational Rose can help in the process of documenting the high-level architecture and use cases.

2.4 Products

The key products of the requirements definition phase are the establishment of evaluation team and the requirements definition document.

Requirements Definition Document (RDD)

The RDD lists the high-level requirements, defines the overall system architecture and its operational environment, and describes how the system will operate within this environment. The document provides a base from which evaluators will select the COTS software. Therefore RDD contains a complete list of all requirements — including low-level, derived requirements — and provides the basis for COTS software selection criteria definition. The format recommended for the document is shown below.

REQUIREMENTS DEFINITION DOCUMENT

1. Introduction

a. Purpose and background of the system

b. Document organization

2. System overview

a. Overall system concept

b. System overview with high-level diagrams showing external interfaces and data flow

c. Discussion and diagrams showing an ideal, high-level architecture for the system

3. Operational environment — description and high-level diagrams of the environment in which the system will be operated

a. Overview of operating scenarios

b. Description and high-level diagrams of the system configuration (hardware and software)

c. Description of the responsibilities of the operations personnel

4. Requirements — functional, operational (interface, resource, performance, reliability, safety, security), and data requirements

a. Numbered list of high-level requirements with their respective derived

2.5 Reviews

At the end of the phase, a requirements definition review is held with management to obtain approval for the plan of attack. The team should highlight the high-level requirements and architecture. They should assess organisational issues such as motivation and politics and how they can impact on the proposed system. The material to be covered at the review meeting is shown below.

REQUIREMENT DEFINITION REVIEW

Introduction — background of the project and system objectives

High-level requirements — overview of requirements, highlighting any changes needed to maximize existing package use

Operational scenarios — use-cases for each requirement or groups of requirements

Strawman system architecture — high-level diagrams of the software system showing interfaces between existing packages and custom-developed components; description or diagram of hardware to be used; and alternatives

Requirements traceability matrix — mapping of requirements to use-cases and system components (packages)

3. SOCIAL-TECHNICAL CRITERIA DEFINITION

Social-technical criteria definition Highlights	
Input:	Customer requirements Requirements Definition document (RDD)
Output:	Evaluation criteria database Evaluation criteria document
Checkpoint:	Priority hierarchy criteria matrix
Steps:	1. Develop a database of the evaluation criteria 2. Define the social-technical criteria 3. Develop the priority hierarchy criteria matrix

3.1 Overview

The purpose of the social-technical criteria definition is to derive attributes or parameters against which the COTS product is evaluated and upon which selection decisions are made. The social-technical criteria include: 1) technology factors, 2) functionality characteristics, 3) product quality characteristics, and 4) social-economic factors. During this phase evaluation attributes for the system are derived from requirements definition document and stakeholders. These should then be revised to incorporate feedback from alternatives identification and evaluation processes. Another possible source of information in defining the criteria is experience from past evaluation cases.

The next step is to build the priority ranking structure or hierarchy matrix for the evaluation criteria showing how a particular evaluation attribute is ranked in relation to other attributes. This is part of the AHP and also will enable management to review the priority of the elicited requirements. At the end of the phase, a criteria definition review is held with management to obtain approval. The evaluation criteria database and evaluation criteria documents are the products of the criteria definition phase.

3.2 Key Activities

Develop a database of the social-technical criteria. The database structure and some of the data (records) can be re-used from previous evaluation cases. However if developed from scratch, then structure of the database should be developed based on the template shown in table 1. The use of template will help the evaluation team to be more focussed.

Define the evaluation criteria for COTS product underlying technology. Derive the criteria for selecting the underlying technology from high-level requirements. A technology is a specification or framework that provides the basis for integrating components for example the customers could specify that they prefer a system that is based on CORBA technology.

List the important underlying technology attributes to which the COTS product must conform? Use AHP or other methods prioritise these attributes according the order of importance using scale 1 to 9 (1=Not preferred and 9= Strongly preferred).

Attribute Name **Priority ranking value**

1.

2.

Define the compliance issues (functionality) attributes for COTS products. Derive the COTS product functionality characteristics from high-level customer requirements. The functionality characteristics help in the initial selection of alternatives. For

List the important functionality attributes to which the COTS product must conform? Use AHP or other methods prioritise these attributes according the order of importance.

Attribute Name **Priority ranking value**

1.

2.

example a user may specify that they want a system that will enable them conduct business on the Internet.

Define the product quality characteristics for COTS product. Derive the product quality characteristics from high-level customer requirements. These are behavioural properties that the product must have and should match the customer's non-functional

List the important quality attributes to which the COTS product must conform? Use AHP or other methods prioritise these attributes according the order of importance

Attribute Name **Priority ranking value**

1.

2.

requirements, for example reliability and portability.

Define the social-economic criteria for COTS product. Derive socio-economic factors from high-level customer requirements and stakeholders. Socio-economic factors are non-technical factors that should be included in the evaluation and selection of COTS components such as costs, business issues, vendor performance and reliability. These

List the important social-economic (non-technical) attributes to which the COTS product must conform? Use AHP or other methods prioritise these attributes according the order of importance.

Attribute Name **Priority ranking value**

1.

2.

are most frequently overlooked factors that bring in the social dimension in the evaluation criteria. The recommended techniques are JAD session and interviews.

Build a priority structure for functionality, quality attributes, and social-economic factors. It is important that a hierarchy of priority is developed for the defined criteria using AHP method. The priority hierarchy should initially be developed for the essential attributes not for the "nice to have" attributes. A brainstorming technique with stakeholders within a JAD session is recommended to build a priority hierarchy.

Using AHP or other methods list functionality, quality attribute and social-economic attributes according the order of importance using scale 1 to 9 (1=Not preferred and 9= Strongly preferred). **Attribute Name Priority ranking value** **1.**

3.3 Methods and Tools

Analytic Hierarchy Process (AHP)

AHP technique is based on pair-wise comparison between the alternatives. The result of this pair-wise comparison is converted to a normalised ranking by calculating the eigenvector from the comparison matrix's largest eigenvalue. The advantages of the AHP technique are that it provides a systematic approach for consolidating information about alternatives using multiple-criteria, it is an objective weighing technique for setting the weighing scale for qualitative and quantitative data, and allows for consistency checking. AHP is fully documented in Saaty (1990).

An example is provided in section 6 of this workbook on how to use AHP to calculate the eigenvector and rank alternatives according to preferences. A tool is also available called expert choice to assist in the AHP process at URL http://www.expertchoice.com/

Joint Application Development (JAD) workshop

Refer to the section 2.3 of this appendix.

Interviews and qualitative data

The most fundamental way of collecting qualitative data is by an in-depth interview. Interview is a face-to-face (telephone), interpersonal role situation in which an interviewer asks respondents questions designed to elicit answers pertinent to the research question.

3.4 Products

The key products of the criteria definition phase are the evaluation criteria database and evaluation criteria document.

Evaluation criteria database

The database contains data on the evaluation criteria to be used in the later stages of the evaluation. The database will include data on the criteria definition, criteria rationale, criteria source of data and criteria ranking priority (refer to table 1 for details). The database provides a base from which the team can use to identify and assess the suitability of the off-the-shelf software. This database is produced by the team as the key product of the criteria definition phase and will act as a basis for reuse for future evaluation cases.

Table 1. Template for criteria definition (available in version 2)

Item name	***Description***
Heading	Heading for each evaluation attribute acts as a unique identifier.
Definition	A definition of the evaluation attribute.
Rationale	Description of the rationale for the evaluation attribute and how it relates to the evaluation criteria.
Scale	The scale or type of description used. *Free format description:* The evaluation attribute will be documented with a free format description. *List:* A list of features, characteristics, functions etc. Is produced. *Structured description:* There is a template or a checklist that defines what should be described for each alternative. *Nominal:* Classes are identified but they are not ordered. *Ordered:* Classes are identified and they are ordered. *Interval:* The scale has meaningful interpretation of distance between entities, but their ratios cannot be calculated, i.e., “there is no meaningful zero point”. *Ratio:* Entities can have ratios, “zero is a meaningful concept”. *Absolute:* The number of entities is counted.
Unit/classes	Definition of the unit of measure or the classes used, which ever is applicable.
Screening rule	Definition of a possible level that is required for an alternative to be selected for detailed evaluation. This field is used for documenting which criteria were used in the screening phase.
Baseline	Baseline is the minimum required level of functionality and features that the application must satisfy when it is delivered.
Qualitative description	Guidelines how additional information about the evaluation attribute should be documented.
Source	How the value for the evaluation attributes can be determined for each alternative.
Priority	Description of how important the particular evaluation attribute is. The priority classes are as follows: *Required:* The value for the evaluation attribute is essential for the evaluation and must be obtained. *Recommended:* It is recommended that the value for the evaluation attribute is obtained, if time available for the evaluation allows it. *Optional:* The result of the evaluation attribute could be useful in the evaluation. The value should be obtained only if all other criteria have been covered and there is time available.

Evaluation criteria document

The evaluation criteria document lists the social-technical evaluation criteria and provides the basis for COTS software selection. The document should include the traceability matrix and the priority hierarchy matrix. The traceability matrix maps the evaluation criteria to the high-level requirements and shows how each high-level requirement or use case is addressed by the social-technical criteria. The priority hierarchy matrix shows how the particular evaluation attribute is ranked in relation to other attributes. The format recommended for the document is shown below.

Introduction - Background to the project and the purpose of the system **Requirements** - High-level requirements of the system. **Evaluation criteria** - List of social-technical requirements i.e. functionality, technology, quality and social-economic. **Traceability matrix** - Mapping of evaluation criteria to requirements.

3.5 Review

At the end of the phase, a criteria definition review is held with management to obtain approval. The team should highlight the priority hierarchy matrix and the rationale for the ranking the evaluation attributes. Issues and problems beyond the control of the team should also be discussed at this meeting.

4. COTS PRODUCT IDENTIFICATION

COTS Product Identification Highlights	
Input:	Social-technical criteria Marketplace
Output:	Component Identification Report Demonstration copies
Checkpoint:	Component Identification Report
Steps:	1. Search for available COTS software components 2. Revise requirements and evaluation criteria 3. Obtain demonstration copies and documentation

4.1 Overview

The objective of this process is to identify COTS components that meet the high level customer requirements so that they can be considered for a more rigorous evaluation. The domains relevant to the problem are identified and available packages in those domains understood. The search for candidate COTS components is conducted through a market survey; Internet search; vendor publications and sales promotions; and computer fairs and shows. The identified components are screened to reduce them to reasonable number so that they can be evaluated in details. The next step in this identification process is to obtain information about the COTS products or obtain the COTS products or both. This information may consist of evaluations done by independent evaluators, reports from vendors, a demonstration of the product capabilities by the vendor and information obtained directly from actual users. The adopted search techniques, customer participation and availability of products have an impact on the success of this process and the whole COTS selection process. At the end of the phase, a component identification report is produced and reviewed with management. In addition demonstration copies of the selected software components are obtained with supporting documentation.

4.2 Key Activities

Select the underlying technology or other keystone issues. Select the underlying technology from the defined criteria for example Distributed Component Object Model (DCOM) from Microsoft Corporation. This will involve assessing and selecting the "best" technology among alternatives and this process can be augmented by reviewing literature on component technology such as component software by (Szyperski, 1998). Understanding and evaluation of the underlying assumptions about the technology must be elicited and analysed before commitment to a particular technology.

Which underlying technology or middleware have you selected for integrating COTS components and why was it selected. Alternatively you could provide a priority hierarchy matrix

Derive the main search and screening criteria. The main search and screening criteria must be based on essential requirements and not optional or "nice to have" requirements. The functionality attribute with the highest or second highest priority from priority hierarchy matrix should be used as the main search and screening criteria, for example a tool that support development of applications in Java.

Search and screen for available COTS products. Search the marketplace to identify candidate COTS products through market surveys and other techniques like Internet search. The search criteria at this stage should be limited to functionality issues and underlying technology. The identified components should then be screened to reduce them to reasonable number so that they can be evaluated in detail in the next phase. This can be effectively achieved by review of documentation on the identified COTS software. Elimination by aspect is for reducing the number of alternative to three or four.

Provide a list of identified candidate products (alternatives). Using elimination by aspect reduce the number of candidate alternative to three (see example in section 6)

Alternatives	**FUN1**	**FUN2**	**FUN3**	**FUN4**	**FUN5**	**FUN6**	**FUN7**	**FUN8**
1.								
2.								
3.								

Order of Aspect	**Remaining alternatives**

List the selected products
1.

Revise requirements and social-technical criteria based on available COTS products. Examine the screened components whether they can handle the requirements or use-cases for the system. Revise the social-technical criteria and update the use-cases based on available components. The selection of the "best" among the packages available depends on the assessment of their compatibility with the requirements specification and the prioritisation of these requirements.

Obtain demonstration copies of selected products and support documentation. Obtain demonstration copies and supporting documentation of the screen products. Demonstration copies can be obtained via the Internet or contacting the vendor directly.

4.3 Methods and Tools

"Word of mouth" and user community

In this method individuals of the evaluation team will draw from experience and networking capability in order to identify available COTS products in the marketplace. The mailing list and product user community can used to elicit more information on available products and their capabilities. This method is recommended because it does not require substantial financial investments.

Computer fairs and shows

Members of evaluation teams can arrange to attend computer fairs and shows in order to identify what COTS products are available in a particular domain. This method depend on the financial capabilities of the vendors to host computer fairs

Internet search

Internet search using search engines is one of the most effective methods for identifying available COTS products in the marketplace. In some cases vendor male available demonstration copies on the Internet which can be downloaded and evaluated. The following are some suggestions for effective searching of the Web using current tools:

(a) General and popular Information. For popular information, directories such as Yahoo! are often useful (e.g. searching for "NEC Research" provides the link to the NEC Research Institute homepage as the only page returned. For more up-to-date information, or for comprehensive results, the major Web search engines (e.g AltaVista, Excite, HotBot, Infoseek, Lycos and Northern Light) are usually preferred.

(b) Harder to find Information. For hard to find information it is possible to obtain more comprehensive results, or have a better chance of finding less popular documents, by combining the results of multiple search engines using a metasearch engine (or simply repeating a search at multiple engines). A good example of a metasearch engine is (http://www.metacrawler.com).

(c) Too many results? Poor relevance. A common complaint against search engines is that they return too many pages, and that many of the pages have low relevance to the query. One of the main problems is that the search engines do not rank the relevance of results very well. For queries returning many hits, search engines like Google (http://google.stanford.edu) and Direct Hit (http://www.directhit.com/) can be useful because they often rank the results better than traditional search engines.

(d) (d)Specialized search engines. More comprehensive and more relevant results may be possible using a search engine that specializes in a particular area. For example Agora (http://agora.sei.cmu.edu) is specialized search engine for software components. "Softbot" search Services such as AHOY! can find pages that are not indexed by any engine. AHOY! is a specialized search service for homepages. It uses metasearch and also adds additional intelligence.

Provide a list of contacts, Internet search engines, product catalogues, special vendors that you use to search for candidate products. **Mailing list** **Reference (or URL)** **1.** **2.** **Internet Search Engines** **Reference (or URL)** **1.** **2.** **Product catalogues** **Reference (or URL)** **1.** **2.**

Market surveys
A market survey can be made using questionnaires in order to identify available COTS products in the market. These methods pre-suppose the availability of vendor mailing list or address book.

Invitation to tender (ITT) or request for proposals (RFP)
Most public institutions use the tender procedures as a method to identify the available products and their capabilities. An advert is placed on the Internet or newspapers and organisations are requested to bid for the tender.

Vendor promotions and publications
Search can be made through vendor publications and catalogues in order to identify available products.

Paper evaluation/ Study documentation
This is the process of evaluating the COTS products based on documentation rather than experimentation. The disadvantage is that the selected products might not interoperate with each other.

Elimination by Aspect
Purpose to reduce the number of candidate products to a fewer manageable number so that they can be evaluated in detail using AHP. Elimination by aspect examines one attribute at a time, making comparisons among alternatives. It eliminates alternatives that do not satisfy some standard and it continues until all alternatives except three or four have been eliminated. The elimination is based on the most important attributes and also those attributes that would eliminate the most alternatives.

4.4 Products

The key product of the alternatives identification phase is component identification report and demonstration copies of the selected components.

Component identification report

The component identification report lists the components considered for screening and the top two or three top components to be considered for detailed evaluation. The report includes any changes to the requirements and the evaluation criteria due to available of components in the marketplace. The format recommended for the document is shown below.

> **Introduction** - Background to the project and system objectives
> **Component requirements** - Type of components needed, main functionality
> **Screened components** - List of components considered and screening criteria
> **Selected components** - Brief summary of two or three top components to be considered for evaluation
> **Requirements changes** - List of changes to the requirements and the evaluation criteria due to available of components in the marketplace.
> **Risk analysis —** discussion of each major risk and mitigation strategy
> **Project estimates —** preliminary cost and schedule estimates

Demonstration copies

These are demonstration copies of the selected components for detailed evaluation. Technical documentation and user manuals accompany these.

4.5 Review

The team should highlight any requirements changes needed to allow the system to be built from available packages. They should provide an analysis of the reasons for the suggested requirements changes, including a discussion of the results of the package screening. The team should assess the impact on the business process of changing or eliminating requirements. They should also provide an estimate of the cost of developing custom software to meet the requirements that cannot be met by any existing package, and they should present the costs and risks of alternative strategies for meeting the requirements (such as making modifications to an acquired package).

5. ASSESSMENT (EVALUATION)

Assessment (Evaluation) and Selection Highlights	
Input:	Social-technical criteria Demonstration copies of selected components
Output:	Evaluation report
Checkpoint:	Evaluation report - executive summary
Steps:	1. Assess the selected components 2. Analyse data and recommend component

5.1 Overview

The evaluation involves contacting vendor technical support for evaluation information, review of vendor documentation and product testing for quality and functionality. It includes evaluating COTS performance, interfaces and ease of integration, comparing short-term and long-term licensing costs against integration costs. The reasons for selecting each component and the reasons for rejecting others should be recorded. The data collection methods are separated from the data analysis method. There are a number of data collection techniques such as examining the products and vendor supplied documentation, viewing demonstration and interviewing demonstrators, executing test cases and applying the products in pilot projects. STACE proposes selecting appropriate techniques depending on resources and experience. In addition data collection may include interviewing actual users of the products, and examining sample outputs from projects that have used the products. AHP should be used for data analysis of the evaluation results.

5.2 Key Activities

Assess products against defined functionality criteria. It is important to thoroughly evaluate each product selected from the screening process in order to select the "best" product among alternatives. The evaluation may involve further review of product documentation, testing the defined functionality criteria and experimentation within operating environment. Then use AHP to do pairwise comparison between the products regarding functionality.

Provide a summary of what each product does (functionality, quality and social-economic factors).

Product A

Product B

Assess products against defined quality characteristics. This activity will involve further testing of products for quality characteristics. This second round of package evaluation focuses on obtaining and analyzing real experience data about the products. Annex 2 of this appendix provides examples of important quality characteristics of COTS software. Then use AHP to do pairwise comparison between the products regarding quality characteristics.

Assess products against defined social-economic criteria. Most of social-economic data can not be easily elicited by experimentation with the products but require interview vendors and other stakeholders to obtain qualitative information. It is important for example that the product is acceptable within the context in which it will be used. These interviews should be extended to other users of the systems to elicit their experiences with the products. Annex 2 of this appendix provides examples of important social-economic factors of COTS software. Then use AHP to do pairwise comparison between the products regarding social-economic factors.

Analyse the data and select product. UseAHP to consolidate the evaluation data in order to select the "best" components among alternatives. There are tools available to support the AHP techniques such as expertchoice. Record the reasons for selecting each component and the reasons for rejecting others. In the event that none of the COTS product fully satisfies the requirements either the process can be repeated or a recommendation can be made to develop a bespoke system.

Complete the priority ranking matrix (refer to case study in section 6)

	Priority Ranking	**Product A**	**Product B**	**Product C**
Functionality characteristics				
Product quality attributes				
Social-economic factors				
Overall ranking				

Activity 5.2

5.3 Methods and tools

Analytic Hierarchy Process (AHP)
Refer to section 3.3 of this appendix.

Demonstrations
This may involve the team attending vendor demonstration of the product. At this meeting the team will prepare some questions and test cases for the vendor.

Experimentation
This is a rigorous test of the product by the team to test for compliance in accordance to the defined criteria. It is important that the experimentation is conducted in the operating environment and the product is tested with real data. It will require generating test cases to help in the evaluation process.

Pilot testing
A pilot study is an extended period of experimentation and testing which you use real data from the organisation. The pilot testing can also be some form of phased approach whereby if the pilot phase work you can extend it to the other phases.

Business analysis
Business analysis techniques are used to evaluate alternative business architectures such as supply chain, option analysis, market position, market potential and saturation, supplier financial indicators.

Audit development process
In this method the team is looking at the development process that was used to develop the software including the tests carried out, conformance to standards, etc. This is important especially for safety-critical systems.

Use of templates, checklists and questionnaires
Use of templates, checklists and questionnaires can assist to streamline the evaluation process. A template and questionnaire is attached as appendix to this report.

5.4 Products

The key product of the assessment phase is the evaluation report.

Evaluation report

The evaluation report presents the evaluation method used and results. This includes a brief summary of the recommended components and why it was selected. The format recommended for the document is shown in below. The evaluation team produces this document as the key product of the assessment phase. The document is distributed to all stakeholders for review and acceptance.

Executive summary - Summary of the requirements and recommended product
Introduction - Background to the project and system objectives
Evaluation process - Brief summary of the evaluation method used
Evaluation results - Presentation of the evaluation results
Recommended component - Brief summary of the recommended components

5.5 Review

At the end of the phase, a review is held with management to present the final report and get management feedback. In the event that management rejects the recommended product.

Provide an Executive summary of the selection process (full page provided)

6. EXAMPLE: SELECTING A WORD PROCESSING SOFTWARE

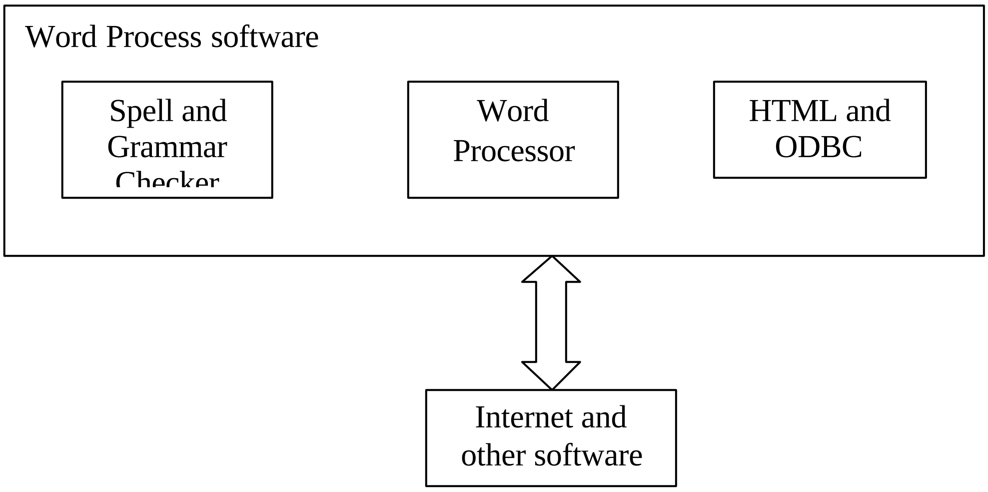
6.1 Problem/ Requirements Definition

Problem definition

ABC management consulting company has just installed Windows NT network for the 10 users. These are for the managing consultant, four consultants, and five support staff including secretaries. The company has been using Word Perfect for windows and most of their clients use either Word Perfect or MS Word. In addition, the company has two laptops with Windows 95 operating system available for use by consultants. The company would like to purchase new word processing software to support their work and writing reports for their clients.

System overview

The company would like to purchase new word processing software to support their work and writing reports for their clients. The word processing software shall support creation, editing, saving and printing of documents. The software shall support HTML and ODBC connectivity. In addition the word processor shall support spell and grammar checking.



Operational environment

The system shall be installed on Windows NT network currently supporting 10 users with potential for growth. The software shall also be installed on two laptops with Windows 95 operating system. The minimum specification of all the machine are intel processor running at 233MHz, 16MB ram and 2GB of hard disk. An external CD driver is available in case the software comes on a CD.

The personnel to operate the systems are the managing consultant, four consultants, and five support staff including secretaries. The company has been using Word Perfect for windows and most of their clients use either Word Perfect or MS Word. Two secretaries and one consultant are not familiar with software on Windows platform having come from Unix background.

Requirements

The system shall support creation, editing, saving and printing of documents. The detailed requirements are presented in below:

(a) The system shall *support multiple file formats*. The system must allow saving documents in other file formats in other programs. For example, open a WordPerfect document, make changes to it and then save it in either MS Word or WordPerfect format. The system shall support automatic save of documents for recovery in case the program hangs (stops responding) or power is lost.

(b) The system shall provide key features that make it faster and easier to proof read and edit documents including the *spell checker and grammar checker*. The system shall check for spelling and grammar errors as text is typed, and highlight possible errors directly in the document. The automatic spelling checker and grammar checker shall offer suggestions for corrections in-place when you right-click the marked text. The system shall support spelling and grammar checking combination when the document is finished.

(c) The system shall provide features to make it faster and easier to work with *tables, borders, and shading.* The system shall provide set of *drawing and graphics capabilities* that you can use to easily embellish your text and graphics with 3-D effects, shadow effects, textured and transparent fills, and AutoShapes.

(e) The system shall provide an extensive set of features that you can use to take advantage of the *World Wide Web and the Internet*. For example the system shall be used to author and browse through rich webs of documents on an intranet or on the Web.

(f) The system shall link to any HTML, or other file on any internal or external Web site or any file server. The system shall automatically recognise and format e-mail addresses, URLs, and UNC path names as hyperlinks. The system shall support sound, video, picture, scrolling text, bullets, horizontal lines, HTML forms and HTML tags. The system shall provide WYSIWYG ("what you see is what you get") support for authoring Web pages with commonly used tags, such as tables, fonts, and background sound.

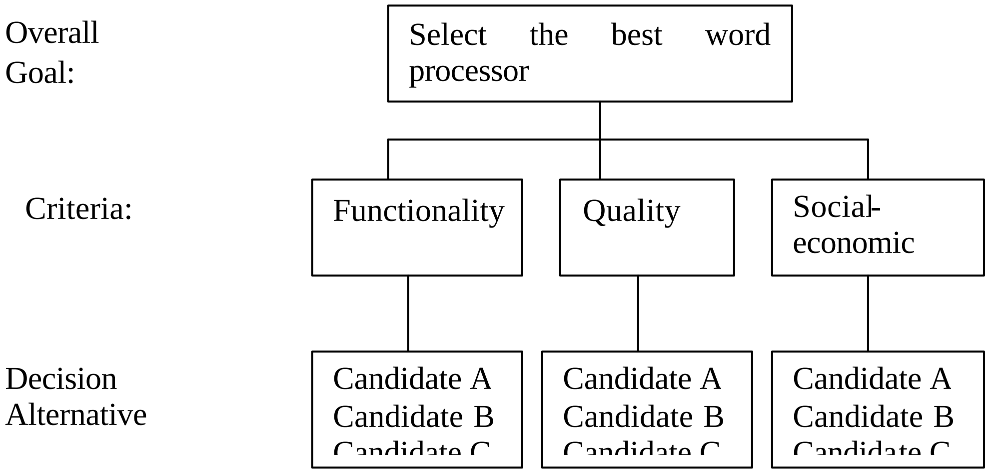
(g) The system shall *support people working in teams* in such that the system versioning and merging documents. Versioning means that the system shall maintain a working history of a document. Merging documents means the system shall consolidate all changes and comments from different reviewers in one easy step. Multiple reviewers can modify separate copies of the same document, and then you can merge all their changes into the original.

(h) The system shall support automatic font and language switching when the keyboard changed, allowing *multilingual text* in your document and in several dialog boxes. Keyboard switching should automatically switch the fonts in your document to appropriate fonts based on the language use when the keyboard is switched. Multilingual text support in dialog boxes shall enable the word processor to edit and display multilingual text across localized versions of word processor.

(h) The system shall *support HTML, SGML (Standard Generalized Markup Language) and ODBC* (Object Database Connectivity).

6.2 Developing the social-technical criteria

Problem Hierarchy



Determining the relative importance of the criteria using AHP

Using pairwise comparisons, the relative importance of one criterion over another can be expressed as follows:

1=equal importance,
3=moderate,
5=strong,
7=very strong,
9=extreme important

The following pair wise comparisons were made to calculate the AHP's eigen vector values:

1. Are functionality characteristics more important than product quality attributes and how much more important with respect to the selection of the word processing software? Answer yes by 5
2. Are functionality characteristics more important than social-economic factors and how much more important with respect to the selection of the word processing software? Answer yes, by 3
3. Are product quality attributes more important than social-economic factors and how much more important with respect to the selection of the word processing software? Answer no, social-economic are more important by 2

	Functionality attributes	*Quality attributes*	*Social-economic factors*
Functionality attributes	*1*	*5*	*3*
Quality attributes	*1/5*	*1*	*1/2*
Social-economic factors	*1/3*	*2*	*1*

You then use mathematically proved eigenvector to turn this matrix into a ranking of criteria. This process involve the following steps:

1. a short computational way to obtain this ranking is to raise the pairwise matrix
2. to powers that are successively squared each time the row sums are then calculated and normalised
3. the computer is instructed to stop when the difference between these sums in two consecutive calculations is smaller than a prescribed value.

For now, let us remove the names and convert the fractions to decimals and then square the matrix:

$$\begin{bmatrix} 1.0000 & 5.0000 & 3.0000 \\ 0.2000 & 1.0000 & 0.5000 \\ 0.3333 & 2.0000 & 1.0000 \end{bmatrix}$$

Multiply by

$$\begin{bmatrix} 1.0000 & 5.0000 & 3.0000 \\ 0.2000 & 1.0000 & 0.5000 \\ 0.3333 & 2.0000 & 1.0000 \end{bmatrix}$$

That is (1.0000*1.0000)+(5.0000*0.2000)+(3.0000*0.3333)= 3.0000 which is the value in the first column and first row.

And (1.0000*5.0000)+(5.0000*1.0000)+(3.0000*2.0000)= 16.0000 first row and second column. Algebra continues until you complete the vector.

$$\begin{bmatrix} 3.0000 & 16.000 & 8.5000 \\ 0.5667 & 3.0000 & 2.1000 \\ 1.0667 & 5.6665 & 3.0000 \end{bmatrix}$$

Now we compute our first eigenvector (to four decimal places). First we sum the rows We sum the row totals and then we normalise by dividing the row sum by the row

$$\begin{bmatrix} 3.0000 + 16.0000 + 8.5000 \\ 0.5667 + 3.0000 + 2.1000 \\ 1.0667 + 5.6665 + 3.0000 \end{bmatrix} \begin{matrix} =27.5000 \\ = 5.6667 \\ = 9.7332 \\ \hline 42.8999 \end{matrix} \qquad \begin{matrix} 0.6410 \\ 0.1321 \\ 0.2269 \\ \hline 1.0000 \end{matrix}$$

totals (I.e. 27.5000 divided by 42.8999 equals 0.6410). The result is our eigenvector. You can then compute the second eigenvector by squaring again the new matrix that we obtain so that we get the second eigenvector. We then compute the differences between the first and second eigenvector. This process must be iterated until the eigenvector solution does not change from the previous iteration. A support tool is available to do this iteration. After these iteration the priority hierarchy matrix below is produced.

Priority hierarchy matrix for the attributes

Criteria	*Priority ranking*	*Comments*
Functionality characteristics	0.6480	Most important criterion
Product quality attributes	0.1220	Least important criterion
Social-economic factors	0.2300	Second most important criterion

The following criteria were derived from this phase and the details about each criterion are stored in the database.

Functionality list	*Product quality list*	*Social-economic*

• Multiple file formats • Spell and grammar checker • Tables, borders, and shading. • Drawing and graphics capabilities • World Wide Web and the Internet. • People working in teams • Multilingual text. • HTML, SGML and ODBC	• Interoperability • Efficiency/ Resource utilisation • Usability	• Costs in general • Licensing arrangements • Vendor reputation • Customer experience

Technology list
Run on Microsoft NT and Windows 95. Please note that the technology criteria is not used in the hierarchy priority because of keystone approach where all the software to be selected must be compatible with the keystone in this case Microsoft NT and Windows 95

Priority hierarchy criteria matrix
Using the AHP the following priority ranking is calculated for each sub criteria.

Main criteria	Sub criteria	Priority ranking
Functionality (I.R.=0.11)	Multiple file formats	0.144
	Spell and grammar checker	0.129
	Tables, borders, and shading.	0.068
	Drawing and graphics capabilities	0.051
	World Wide Web and the Internet.	0.179
	People working in teams	0.211
	Multilingual text	0.051
	HTML, SGML and ODBC	0.166
Product quality (I.R.=0.02)	Interoperability	0.238
	Efficiency/ Resource utilisation	0.136
	Usability	0.625
Social-economic (I.R.=0.04)	Costs in general	0.487
	Licensing arrangements	0.303
	Vendor reputation	0.071
	Customer experience	0.139

I.R. = Inconsistency ratio and note that an inconsistency ratio of 0.1 or greater may warrant investigation.

6.3 Identification of candidate products

The identification of candidate word processing was based on Internet search and six products were selected. A paper evaluation was done and feature analysis list was developed shown in the table below. Elimination by aspect method was used to reduce the alternatives to three for further analysis. Elimination by aspect examines one attribute at a time, making comparisons among alternatives. It eliminates alternatives that do not satisfy some standard and it continues until all alternatives except three or four have been eliminated. The elimination is based on the most important attributes and also those attributes that would eliminate the most alternatives.

Products	FUN1	FUN2	FUN3	FUN4	FUN5	FUN6	FUN7	FUN8
A1	Y	N	Y	Y	N	Y	N	N
A2	Y	Y	Y	Y	Y	Y	Y	Y
A3	Y	Y	Y	N	Y	Y	Y	Y
A4	Y	Y	Y	Y	Y	Y	Y	N
A5	N	Y	Y	Y	Y	Y	Y	Y
A6	Y	Y	Y	Y	Y	Y	N	Y

NOTE: FUN1=Multiple file formats; FUN2=Spell and grammar checker; FUN3=Tables, borders, and shading; FUN4=Drawing and graphics capabilities; FUN5=World Wide Web and the Internet; FUN6=People working in teams; FUN7=Multilingual text; FUN8=HTML, SGML and ODBC

The customer's order of preference according to AHP priority hierarchy was FUN6 (people working in teams), FUN5 (WWW and Internet), FUN8 (HTML, SGML and ODBC), FUN1 (multiple file formats), and so on. The selection process is depicted below, where alternatives that do not possess the required aspect are crossed out.

Order of aspect	Remaining alternatives					
FUN6	A1	A2	A3	A4	A5	A6
FUN5	~~A1~~	A2	A3	A4	A5	A6
FUN8		A2	A3	~~A4~~	A5	A6
FUN1		A2	A3		~~A5~~	A6

The three remaining alternatives A2= Microsoft Office (Word), alternative A3=Lotus SmartSuite (WordPro) and alternative A6=Corel Office Professional (WordPerfect). This will be evaluated in detail later using experimentation and AHP.

6.4 Assessment

Having experimented with the software the following information was discovered about each product.

Microsoft Office97 (c*ost = £523)*

Microsoft Word has an easy-to-use interface with a range of toolbars or icon bars for specific tasks. Word has good zoom controls and it is easier to split and adjust the sizes of the document windows. In Word formatting is speeded up with a painter that allows you to pickup an existing format and then apply it to other areas of the document. Regarding drawing graphics, Word is easy to use and very flexible, for example lines, boxes and circles can be drawn easily anywhere on the page. This makes Word far more suitable for desktop publishing tasks. Word is also capable of importing a wider variety of graphics image formats. Word highlights misspelled or mistyped word and right-clicking on a misspelled word pops up automatically a list of alternatives. Word supports auto correction. Creating tables is relatively simple in Word by selecting the number of rows and columns needed from the drop-down table icon. Word allows for autoformats to tables, which gives a document a professional look. Word is good at dealing with long documents and there are facilities for creating tables of contents, indexes and cross-references. Word helps to create and modify web pages, either starting entirely from scratch or by basing them on existing documents. Word also provides a pop-up assistant that analyses what is being done and offers suggestions as to better or simpler ways of doing it.

Lotus SmartSuite97 *(cost = £374)*
Lotus WordPro has an easy-to-use interfaces with a range of toolbars or icon bars for specific tasks. Basic text entry and formatting is pretty much the same as in Word, though Word has much better zoom controls and it is easier to split and adjust the sizes of the document windows. WordPro offers several formatting icons that allow to 'cycle' through options such as font colours, attributes and alignment options. This is quicker than doing the equivalent in Word. Regarding drawing graphics, the drawing elements such as lines, boxes have to be contained within a drawing frame, which makes WordPro less easier to use and less flexible. WordPro highlight mistakes of misspelled or mistyped words. However, it does not allow right-clicking on a misspelled word to pop-up alternatives automatically. Creating tables is relatively also simple in WordPro by selecting the number of rows and columns from the drop-down table icon. In a table cell within WordPro the text can be oriented in four ways including upside down, whereas in Word it restricted to three text orientations. WordPro is also good at dealing with long documents and includes facilities for creating tables of contents, indexes and cross references. WordPro also helps to create and modify web pages, either starting entirely from scratch or by basing them on existing documents but it is not as good as Word for ease of use and flexibility. In terms of market share, WordPro has a great deal of catching up to do.

Corel Office Professional V7 *(Cost = £399)*
Corel Office consists of the WordPerfect word processor, Quattro Pro spreadsheet, Paradox database and Corel presentations business graphics. The entire suite is very resource hungry, and though the software will work in just 8Mb RAM, it is better with at least 16Mb or 32Mb for decent performance. The software support the IBM VoiceType Control system, which allows to control the software by giving voice commands. The WordPerfect module provides all the usual functions of a modern word processing package and is quite capable of being used as a basic desktop publishing package. WordPerfect includes sophisticated formatting, the ability to embed graphics, a spelling checker, thesaurus and automatic generation of indices or tables of contents. Because there are a lot of WordPerfect users out there, Corel has wisely provided an option that makes menus and functions from previous versions work with this one. Corel has included Internet connectivity in all four core applications. This means that you convert documents automatically to HTML format for publishing on the Internet from within WordPerfect or other core applications.

Assessing alternatives against criteria
In terms of quality characteristics, pairwise comparisons are made to determine the preference of each alternative over another. The following pair wise comparisons were made to calculate the AHP's eigen vector values. Regarding interoperability with other products is Lotus's Wordpro better than Microsoft Word and if yes how much better? Regarding interoperability with other products is Lotus's Wordpro better than Corel WordPerfect and if yes how much better? Regarding interoperability with other products is Corel WordPerfect better than Microsoft Word and if yes how much better? This produces the matrix below.

	Lotus SmartSuite	Corel Office Pro	Microsoft Office97
Lotus SmartSuite97	*1*	*1*	*1/2*
Corel Office Pro	*1*	*1*	*1/2*
Microsoft Office97	*2*	*2*	*1*

The eigenvector is calculated from this matrix and this is shown in the first row in the table below. This process is repeated for the Efficiency/Resource utilisation and usability and the whole table is completed

Priority ranking Matrix

	Lotus SmartSuite97	**Corel Office Professional 7**	**Microsoft Office97**
Interoperability	0.250	0.250	0.500
Efficiency/ Resource utilisation	0.582	0.109	0.309
Usability	0.136	0.238	0.625

In to produce the priority ranking for functionality, these values are multiplied by the priority ranking these quality attributes (derived earlier on).

$$\begin{bmatrix} 0.250 & 0.250 & 0.500 \\ 0.582 & 0.109 & 0.309 \\ 0.136 & 0.238 & 0.625 \end{bmatrix} * \begin{bmatrix} 0.136 \\ 0.238 \\ 0.625 \end{bmatrix}$$

The result is priority hierarchy matrix for alternatives regarding quality attributes shown below

Criteria	*Priority ranking*	*Comments*
Lotus SmartSuite97	0.229	Second most preferred alternative
Corel Office Pro	0.224	Least preferred alternative
Microsoft Office97	0.547	Most preferred alternative

This process is repeated for the functionality and socio-economic factors. The priority ranking using AHP is shown in the table below. The table shows that Microsoft Office97 is the recommended package to select. It can be noted from this table that although Corel Office Professional7 (Word Perfect) was a preferred package regarding the social-economic criteria it did not emerge as the winning package because according to ABC company social-economic factors had low priority compared to functionality issues.

Priority ranking Matrix

	Priority Ranking	**Lotus SmartSuite 97**	**Corel Office Professional7**	**Microsoft Office97**
Functionality	0.648	0.267	0.331	0.403

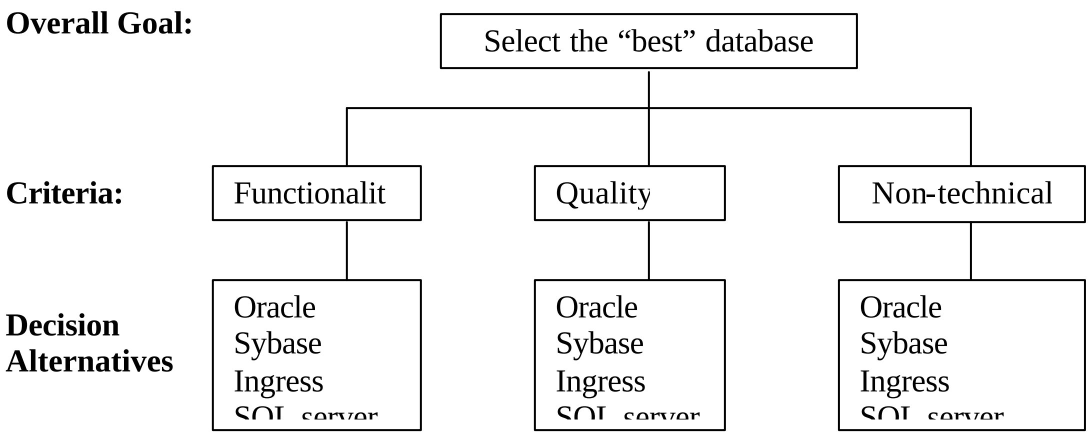
characteristics				
Product quality attributes	0.122	0.224	0.229	0.547
Social-economic factors	0.230	0.332	0.419	0.249
Overall ranking		**0.278**	**0.341**	**0.381**

The overall rating shows that Office97 is a preferred word processing package for ABC Company.

Annex 1. Another example using STACE to select a database (provided with second version of workbook in PowerPoint presentation)

The objective of this example is to show how to select the best commercial database (i.e from Oracle, Sybase, Ingress, SQL server) using AHP.

Step 1. Develop a hierarchical tree of the overall goal, criteria and the decision alternatives.



Step 2. Using pairwise comparisons, the relative importance of one criterion over another can be expressed i.e., 1 equal, 3 moderate, 5 strong, 7 very strong and 9 extreme.

Here's our pairwise matrix with names

	Functionality	Quality	Non-technical
Functionality	1/1	1/2	3/1
Quality	2/1	1/1	4/1
Non-technical	1/3	1/4	1/1

Step 3. Compute the eigenvector for the above pairwise matrix (software tool such as expert choice can be used).

And the computed eigenvector gives us the relative ranking of our criteria

Functionality	0.3196

Quality	0.5584
Non-technical	0.1220

The relative ranking shows that quality aspects are the most important criterion while non-technical are the least important criterion.

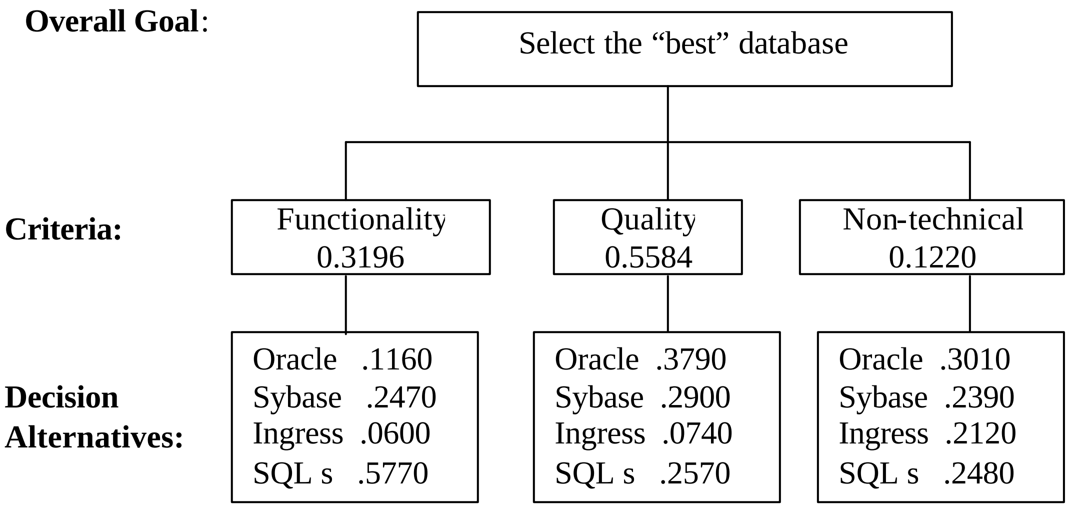
Step 4. Assess the alternatives against criteria

In terms of functionality, pairwise comparisons determines the preference of each alternative over another

	Oracle	Sybase	Ingress	SQL server
Oracle	1/1	1/4	4/1	1/6
Sybase	4/1	1/1	4/1	1/4
Ingress	1/4	1/4	1/1	1/5
SQL server	6/1	4/1	5/1	1/1

This process is repeated for quality and non-technical criteria.

The result is shown in the hierarchical tree with all the weights



Step 5. Consolidate the results of the matrix to obtain the overall ranking

To consolidate the results of the matrix

	Functionality	Quality	Non-technical
Oracle	.1160	.3790	.3010
Sybase	.2470	.2900	.2390
Ingress	.0600	.0740	.2120
SQL server	.5770	.2570	.2480

*

	Criteria
Functionality	0.3196
Quality	0.5584
Non-technical	0.1220

i.e., for Oracle (.1160*.3196)+(.3790*.5584)+.3010*.1220) = .3060

Oracle	.3060
Sybase	.2720
Ingress	.0940
SQL server	.3280

Therefore, SQL server is the highest ranked database

Annex 2. List of social-technical criteria factors

Criteria	Sub-criteria		
Compliance issues (Functionality)	*Need satisfaction measures* Effectiveness Responsiveness Correctness Verifiability Suitability	*Functionality* Customer/Organisations standards Functionality (domain specific) Organisational policies	
Product quality characteristics	*Adaptability* Interoperability Portability Replaceability Reusability Scalability	*Maintainability* Understability	*Performance measures* Dependability Efficiency/ Resource Utilisation Usability
Technology	*Support for integrability* Architectural styles and frameworks Interface issues Support for debug and testing Support for scripting language	Support plug and play Technology standards or protocols *Technology functionality* Multi-user support Ease of migration to other platforms	*Technology quality characteristics* Documentation Technology performance Security issues Technology dependability & resource utilisation
Socio-economic (non-technical)	*Business issues* Contractual (legal) issues Cost of adapting and integrating Costs in general Escrow or buy rights Licensing arrangements Upgrade costs Product costs Cost of operation	Technology costs Training costs *Marketplace variables* Changes in marketplace Delivery period Market trends (viability) Product/technology reputation (maturity, stability) Product/technology	*Vendor capability* Availability of training and support Local support Vendor certification Vendor reputation Vendor stability

	Support costs	restrictions	
Organisational considerations	Allocation of resources Cost justification Customer expectations Customer experience Customer ownership and motivation Customer participation	Customer resistance Incentives Management (sponsor) support Management resistance Management structure Organisational culture	Organisational politics People's attitude Short term approach Skills profile *Evaluation constraints* <u>Benefits and risks</u>

Annex 3. Template for customer experience criteria **<u>(available only in second version)</u>**

Item name	*Description*
Heading	Customer experience
Definition	Customer experience means that the users of the system will be familiar with the system or that they will easily learn the system because they may have used a similar system.
Rationale	It is important that the system procured is familiar with users to reduce the learning curve and cost associated with training.
Scale	The evaluation attribute will be documented with a free format description.
Unit/classes	None
Screening rule	TBD
Baseline	Compatible to WordPerfect and MS Word.
Qualitative description	None
Source	Interviewing the user will used to determine the value for the evaluation attribute.
Priority	*Required*, the value for the evaluation attribute is essential for the evaluation and must be obtained.

15 Appendix 7: Protocol and questions for study 3

The case study protocol is divided into five sections. Section 1 provided the background information, the theoretical framework and the objective of the case study (presented in thesis as part of chapter 7). Section 2 described the key features of the case study method. It presented the case study research design, methodology, data collection and case study database (presented in thesis as part of chapter 5). Section 3 outlined the field procedures (credentials and access to the case study sites, general sources of information and procedural reminders). Section 4 presented the case study questions. Section 5 discussed the case study analysis plan (presented as part of chapter 4) and case study report format.

A. FIELD PROCEDURES

Initial scheduling of field visit

Initial contact was made with organisations that participated in the first study. The field procedures began with verification of access procedures, interview with managers or contact person, review of documentation and conducting interviews to elicit data.

The draft timetable of the field visit is shown in table below.

Activities	**Venue**	**Dates**
Pilot evaluation with one organisation in Zambia	Lusaka, Zambia	14/7/99 - 15/7/99
Case 1		
Verification of access procedures	Lusaka, Zambia	17/8/99 - 20/8/99
Interview Manager/ Contact person	Case study site	23/8/99
Conduct interviews with others	Case study site	24/8/99
Review documentation	Case study site	25/8/99
Draft case study 1 report for comments	Lusaka, Zambia	26/8/99 - 31/8/99
Review of draft report by key informants	Case study site	1/9/99
Final case study 1 report	Lusaka, Zambia	15/9/99 - 20/9/99
Case 2 up 9		
Same activities as case 1	See above	21/9/99 - 20/11/99
Integration		
Cross-case study conclusions	University of York	4/12/99 - 15/12/99
Modify theory	University of York	16/12/99 - 23/12/99
Write cross-case report	University of York	25/12/99 - 30/12/99

Persons to be interviewed and other sources of information

The persons (stakeholders in the COTS evaluation) to be interviewed include policy makers, procurements agents, engineers, system architects, program managers, end users and domain specialists.

The documents to be collected include worked example of using the STACE workbook, written comments in the STACE workbook, evaluation results from the use of the STACE framework and other relevant documents supporting the evaluation of the STACE framework.

B. INTERVIEW QUESTIONS

Contextual Questions

Introduce myself, purpose of the evaluation and thank them for accepting to participate in the evaluation of the STACE framework. The outline of this interview, we will initially discuss your perception of the STACE framework rating in terms gain satisfaction, in the second place we will discuss its rating in terms of interface satisfaction, then quality of life satisfaction and task support satisfaction. Lastly we will conclude our discussion with some problems and recommendations. The interview will not last more than one hour.

(Q1) Before we proceed into the detailed discussion let me begin by asking, what is your responsibility in your organisation?

(Q2) What were you trying to evaluate when you used the STACE framework? Is it possible that I can have a copy of the results of the evaluation in which you used the STACE framework? *(These questions can be asked at the end).*

STACE framework gain satisfaction

3. How did you find STACE with regard to its perceived usefulness? For example, do you find that using STACE method enable you to accomplish tasks more quickly, improve job performance, increase your productivity, enhance your effectiveness on the job, make it easier to do your job and useful in your job
4. How do you rate the method's support for decision-making or decision-making satisfaction? Did the STACE method enable you to make better decisions, lead you to greater use of analytical aids in you decision making and did it make more relevant information available for decision making
5. How would you rate STACE with regard to its appropriateness for task, comparison with alternatives (other available guidance), cost-effectiveness, and clarity (illuminate the process)?

STACE framework interface satisfaction

6. How did you find STACE regarding perceived ease of use? For example, did you find learning to use STACE method easy for you, easy to get STACE method to do what you want it to do, easy for you to be skilful at using STACE method?
7. How do you rate STACE with regard to its presentation, internal consistency, organisation and appropriateness for audience? For example, is the data you need displayed in a readable and understandable form? Is the data presented in a readable and useful format? Did you find it internally consistent? Did you find STACE well-organised and appropriate

STACE framework quality of life satisfaction

8. How did you find STACE in regard to quality of life satisfaction? For example, do you think the framework provide users' feelings of participation?

STACE framework task support satisfaction

9. How did you find STACE with regard to ease of implementation, for example did you find the procedures in STACE easy to implement? Do you think some procedures would require some tailoring, if so which ones?
10. How do you rate STACE with regard to its understandability for example did you find it complex, simple to understand, was it well structured, and readable?
11. How would rate STACE with regard to the completeness of STACE features and procedures were they adequate and sufficient? How about completeness of output information and was it self-contained?
12. To what extent does the framework produce results that you expected? To what extent does this system provide reports or outputs to you that seem to be just about what you need? To what extent do you actually use the reports or outputs by the system

STACE features and principles

13. Which features and principle did you find interesting or useful?
14. Which of the following features and principle of STACE were you satisfied or not satisfied/ (useful or not useful)? Customers participation, Social-technical criteria, Analytical Hierarchy Process (AHP), Keystone evaluation strategy or elimination by Aspect, Techniques for identification of candidate components, Evaluation techniques, Classification scheme

General and conclusion

15. What problems did you encounter with using STACE method? What are the potential problems or limitations of the framework?
16. What areas of improvements would you recommend for STACE framework? What do you think is missing or not clear in the framework?

16 References

African Internet Connectivity (AIS), "Information & Communication Technologies (ICTs) telecommunications, Internet and computer infrastructure in Africa," http://www3.sn. apc.org/africa/, 1999.

Alford M., "Panel session issues in requirements engineering technology transfer: from researcher to entrepreneur," Proceedings of the first International Conference on Requirements Engineering (ICRE'94), IEEE Computer society press, Colorado Springs, Colorado, pp. 144, April 18-22, 1994.

Allen P. and Frost S., "Component-based development for enterprise systems: applying the SELECT perspective," Cambridge University Press, Cambridge, 1998.

Anderson E., "A heuristic for software evaluation and selection," Software Practice and Experience, Vol. 19, No. 8, pp. 707-717, August 1989.

Averweg U. R. and Erwin G. J., "Critical success factors for implementation of decision support systems in South Africa," Proceeding of the 32nd Hawaii International Conference on System Sciences, IEEE computer society, Los Alamitos, California, 1999.

Avgerou C. and Cornford T., "Developing information systems: concepts, issues and practice," 2nd ed., MacMillan Press Limited, 1998.

Avgerou C. and Land F., "Examining the appropriateness of information technology" in Social Implications of Computers in Developing Countries, Bhatnagar S. C. and Odedra-Straub M. (eds.), Tata MacGraw-Hill Publishing Company Limited, New Delhi, pp. 26-41, 1992.

Avgerou C., "Computer-based information systems and modernisation of public administration in developing countries" in Information Technology in Developing Countries, Bhatnagar S.C. and Bjorn-Andersen N. (eds.), Elsevier Science Publishers, North-Holland, pp. 243-250, 1990.

Avgerou C., "Transferability of information technology and organisational practices, in Global information technology and socio-economic development, Odedra-Straub M. (ed.), Ivy League, New Hampshire, pp. 106-115, 1996.

Avison D. E and Fitzgerald G., "Information systems development: methods, techniques and tools," McGraw-Hill Book Company, London, 1995.

Avison D. E. and Wood-Harper A. T., "Multiview: an exploration in information systems development," Blackwell Scientific Publications, London, 1990.

Axtell C. M., Waterson P. E. and Clegg C. W., "Problems integrating user participation into software development," International Journal of Human-Computer Studies, Academic Press Limited, Vol. 47, pp. 323-345, 1997.

Babbie E., "The practice of social research," Wadsworth Publishing Company, Belmont, USA, 1998.

Banner D. K. and Gagne T. E., "Designing Effective Organisations: Traditional and Transformational views," Sage Publications, London, 1995.

Bell F. and Davis R. C., "Using SSM and software prototyping: an emergent methodology for an ethical information system," in Methodologies for Developing and Managing Emerging Technology based Information Systems, Wood-Harper A. T., Jayaratna N. and Wood J. R. G. (eds.), Springer-Verlag London Limited, pp. 1-14, 1999.

Bell S. and Wood-Harper T., "Information Systems Development for developing countries" chapter in Information Technology in Developing Countries, Bhatnagar S.C. and Bjorn-Andersen N. (eds.), Elsevier Science Publishers, North-Holland, pp. 23-37, 1990.

Bell S., "Learning with information systems: Learning cycle in information systems development," Routledge, London, 1996.

Beus-Dukic, L. and Wellings A., "Requirements for a COTS software component: a case study," Conference on European Industrial Requirements Engineering (CEIRE '98), Requirements Engineering, Springer-Verlag, Vol.3, No.2, pp. 115-120, October 1998.

Beynon-Davies P., "Human error and information system failure: the case of the London ambulance service computer-aided despatch system project," Interacting with Computers 11, pp. 699-720, 1999.

Bhatnagar S. C. (ed.), "Information Technology Manpower: Key Issues for Developing Countries," Tata McGraw-Hill Publishing Company Limited, New Delhi, 1992a.

Bhatnagar S. C. (ed.), "Social Implications of Computers in Developing Countries," Tata MacGraw-Hill Publishing Company Limited, New Delhi, 1992b.

Bihari T. E. and Varner M. O. "Practical issues in information technology transfer" in Levine L. (ed.), Diffusion, Transfer and Implementation of Information Technology (A-45), Elsevier Science B.V. (North-Holland), pp. 369-372, 1994.

Bjorn-Andersen N., "Are 'Human factors' Human?" Computer Journal, Vol. 31, No.5, pp. 386-390, 1988.

Bjorn-Andersen N., "Information technology in developing countries - for better for worse" in Bhatnagar S.C. and Bjorn-Andersen N. (eds.), Information Technology in Developing Countries, Elsevier Science Publishers, North-Holland, pp. 279-283, 1990.

Boehm B. and Abts C., "COTS Integration: Plug and Pray?" IEEE Computer, Vol. 32, No. 1, pp. 135-138, January 1999.

Bogod J., "The role of computing in developing countries," British Computer Society, London, 1979.

Boloix G. and Robillard P., "A Software System Evaluation Framework," IEEE Computer, Vol. 28, No. 12, pp. 17-26, December 1995.

Bostrom R. P. and Heinen J.S. "MIS problems and failures: A sociotechnical perspective. Part I: The causes," MIS Quarterly, pp. 17-32, September 1977.

Braun C. L. "A Lifecycle Process for the Effective Reuse of Commercial Off-the-Shelf (COTS) software," Proceeding of the Fifth Symposium on Software Reusability, SSR'99, pp. 29-36, ACM Press, 1999.

Bravo E., "The Hazards of leaving out users," In Participatory Design: Principles and Practices, Schuler D. and Namioka A. (eds.), Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 3-12, 1993.

Brown A. W. and Carney D. J. and McFalls M. D., "Proceedings of the SEI/MCC Symposium on the Use of COTS in Systems Integration," Special Report CMU/SEI-95-SR-007, Pittsburgh, USA, 1995.

Brown A. W. and Short K. "On components and Objects: The foundations of Component-Based Development," in Proceedings of the Fifth International Symposium on Assessment of Software Tools, IEEE Computer Society, Los Alamitos, California, pp. 112-121, June 1997.

Brown A. W. and Wallnau K. C., "A Framework for Systematic Evaluation of Software Technologies," IEEE Software, Vol. 13, No. 5, pp. 39-49, 1996a.

Brown A. W. and Wallnau K. C., "Engineering of Component-Based Systems," in Brown A. W. (ed.), "Component-based Software Engineering: Selected papers from Software Engineering Institute," IEEE Computer Society Press, Los Alamitos, California, pp. 7-15, 1996b.

Bryman A. and Cramer D., "Quantitative data analysis with SPSS for Windows: A guide for social scientists," Routledge, London, 1997.

Calhoun C. and DeLargy P. F., "Computerisation, aid-dependency and admininstrative capacity: a Sudanese case study," in Grant-Lewis S. and Samoff J. (eds.), Microcomputers in African development: Critical perspective, Westview press, pp. 25-63, 1992.

Care N. S., "Participation and policy," Ethics 78 (July), pp. 316-337, 1978.

Carmel E., Whitaker R. D. and George J. F., "PD and Joint Application Design: A transatlantic comparison," Communications of the ACM, Vol. 36, No. 4, pp. 40-48, 1993.

Carnegie Mellon University (CMU), "COTS-Based Systems Initiative," Carnegie Mellon Software Engineering Institute, http://www.sei.cmu.edu/cbs/, 1998.

Carney D. J. and Wallnau K. C., "A basis for evaluation of commercial software," Information and Software Technology, Vol. 40, pp. 851-860, 1998.

Cavage A. L. M., "Case study research: a multi-faceted research approach for IS," Information Systems Journal, Vol. 6, pp. 227-242, 1996.

CCTA, "Managing successful projects with PRINCE 2 (PRINCE2 reference manual)," Stationery Office for CCTA, Norwich, 1998.

Clements P. C., "From Subroutines to Subsystems: Component-Based Software Development," in Brown A. (ed.), Component-Based Software Engineering, IEEE Computer Society press, Los Alamitos, California, pp. 3-6, 1996.

Collins B. A. and De-Diana I. P.F., "Sharing and adapting educational computer software" in Bhatnagar S. C. and Odedra M. (eds.), Social Implications of Computers in Developing Countries,Tata MacGraw-Hill Publishing Company Limited, New Delhi, pp. 319-327, 1992.

Coppit D. and Sullivan K. J., "Multiple mass-market applications as components," Proceedings of the International conference of Software Engineering (ICSE), IEEE computer society, Los Alamitos, California, pp. 273-282, 2000.

Corr P. H., "Computer Education in Least Developed Countries: A Case Study of Zambia" in Heeks R., Bhatt P., Huq M., Lewis C., Shibli A. (eds.), Technology and developing countries: Practical applications, theoretical issues, Frank Cass & Co. Ltd, London, pp. 87-100, 1995.

Cramer D., "Fundamental Statistics for Social Research: step-by-step calculations and computer techniques using SPSS for Windows," Routledge, London, 1998.

CSO, "Living Conditions Monitoring Survey Report," Central Statistical Office, Lusaka, 1998.

Curtis B. and Krasner H. and Iscoe N., "A field study of the software design process for large systems," Communication of the ACM, Vol. 31, No. 11, pp. 1268-1286, November 1988.

Daly J. A., "Foreign assistance agencies as advocates and innovators," in Grant-Lewis S. and Samoff J. (eds.), Microcomputers in African development: Critical perspective, Westview press, pp. 147-184, 1992.

Darke P., Shanks G. and Broadbent M., "Successfully completing case study research: combining rigour, relevance and pragmatism," Information Systems Journal, Vol. 8, pp. 273-289, 1998.

Davidson E. J., "Joint application design (JAD) in practice," The Journal of Systems and Software, Vol. 45, pp. 215-223, 1999.

Davis G. B. and Olson M. H., "Management Information Systems: Conceptual foundations, structure and development," McGraw-hill Publishing Company, New York, 1985.

Dean J. C. and Vigder M. R., "System Implementation Using Commercial Off-The-Shelf Software," Proceedings of the 1997 Software Technology Conference (STC 97), Salt Lake City, Utah, May 1997.

Dyer J. S., "Remarks on the Analytical Hierarchy Process," Management Science, Vol. 36, No. 3, pp. 249-259, 1990.

Easterby-Smith M. and Thorpe R. and Lowe A., "Management Research: An Introduction," Sage Publications Ltd, London, 1991.

Eisenhardt K. M. "Building theories from case study research," Academy of Management Review, Vol. 14, No. 4, pp. 532-550, 1989.

Emam K. E., Quintin S. and Madhavji N. H., "User participation in the Requirements Engineering Process: An Empirical Study" Requirements Engineering, Springer-Verlag London Limited, Vol. 1, No. 1, pp. 4-26, 1996.

Enns H. G. and Huff S. L., "Implementation of Information Technology in developing countries: Experience of a Mongolian Internet Service Provider," Proceeding of the 32nd Hawaii International Conference on System Sciences, IEEE computer society, Los Alamitos, California, 1999.

Fan M., Stallaert J. and Whinston A.B., "The adoption and design methodologies of component-based enterprise systems," European Journal of Information Systems, Vol. 9, pp. 25-35, 2000.

Finkelstein A., Spanoudakis G. and Ryan M., "Software Package Requirements and Procurements," Proceedings 8th International Workshop on Software Specification and Design, IEEE Computer Society Press, pp. 141-145, 1996.

Flynn D., "Information Systems Requirements: Determination & Analysis," McGraw-Hill Publishing Company, Berkshire, England, 1998.

Fowler P., "Working Session 1: Critical Issues in the First to Third World Transfer of Software and Information Technology" in Levine L. (ed.), Diffusion, Transfer and Implementation of Information Technology (A-45), Elsevier Science B.V. (North-Holland), pp. 395-402, 1994.

Fox G., Lantner K. and Marcom S., "A Software Development Process for COTS-based Information System Infrastructure," in Proceedings of the Fifth International Symposium on Assessment of Software Tools, IEEE Computer Society, Los Alamitos, California, pp. 133-142, June 1997.

Fransman M., "Technology and Economic Development," Wheatsheaf, Brighton, 1986.

Friedman B. and Kahn Jr. P. H., "Educating Computer Scientists: Linking the social and technical," Communication of the ACM, Vol. 37 No.1, pp. 65-70, January 1994.

Garrity E. J. and Sanders L. G., "Dimensions of Information Systems success in Garrity E. J. and Sanders L. G. (eds.), Information Systems Success Measurement," Idea Group Publishing (IGP), Hershey, pp. 13-45, 1998.

Gould J. D., Boies S. J. and Lewis C. H., "Making usable, useful, productivity-enhancing computer applications," Communications of the ACM, Vol. 34, pp. 74-85, 1991.

Grant-Lewis S. and Samoff J. (eds.), "Microcomputers in African Development: Critical Perspectives," Westview Press, Boulder, 1992

Gronbaek K., Grudin J., Bodker S. and Bannon L., “Achieving co-operative software design: Shifting from product to a process focus” in Schuler D. and Namioka A. (eds.), Participatory Design: Principles and Practices, Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 79-97, 1993.

Grudin J., “Eight challenges for developers,” Communications of the ACM, Vol. 37, No. 1, pp. 93-105, January 1994.

Gyamfi-Aidoo J., Soko M. C. and Kunda D., “Integration of Environmental Information into development planning and decision making processes,” UNEP/ECZ report No. EAP.MR/95/18, Lusaka, 1995.

Haines G., Carney D. and Foreman J., “Component-Based Software Development/ COTS Integration,” Software Technology Review, Software Engineering Institute, http://www.sei.cmu.edu/str/descriptions/ cbsd_body.html>, 1997.

Harker P. T. and Vargas L. G., “Reply to - Remarks on the Analytical Hierarchy Process – by J. S. Dyer,” Management Science, Vol. 36, No. 3, pp. 269-273, 1990.

Hassan S. Z., “Software industry evolution in a developing country: an in depth study,” Proceeding of the 33rd Hawaii International Conference on System Sciences, IEEE computer society, Los Alamitos, California, pp. 1-10, 2000.

Heeks R., “Software strategies in developing countries,” Communication of ACM, Vol. 42, No. 6, pp. 15-20, 1999.

Heeks R., “Technology and development,” in Heeks R., Bhatt P., Huq M., Lewis C., Shibli A. (eds.), Technology and developing countries: Practical applications, theoretical issues, Frank Cass & Co. Ltd, London, pp. 1-11, 1995.

Hicks J. O., “Management Information Systems: A user perspective,” West Publishing Company, 1993.

Hirschheim R. and Newman M., “Information Systems and User Resistance: Theory and Practice,” Computer Journal, Vol. 31, No. 5, pp. 398-408, 1988.

Hokey M., “Selection of Software: The Analytic Hierarchy Process,” International Journal of Physical Distribution and Logistics Management, Vol. 22, No. 1, pp. 42-52, 1992.

IEEE Std 1209-1992, “IEEE Recommended Practice for the Evaluation and Selection of CASE Tools,” IEEE, New York, 1993.

Ingham B., “Economics and development,” McGraw-Hill book company, London, 1995.

ISO/IEC 9126: 1991, “Information technology-Software product evaluation-Quality characteristics and guidelines for their use,” ISO/IEC, Geneva, 1991.

Ives B. and Olson M. H., “User involvement and MIS success: A review of research,” Management Science, Vol.30, No.5, pp. 586-603, May 1984.

Jacobson I., “The Use-Case Construct in Object-Oriented Software Engineering,” in Carroll J. M. (eds.), Scenario-Based Design: Envisioning Work and Technology in System Development, John Wiley and Sons, pp. 309-336, 1995.

Janczewski L. J., “Factors of Information Technology Implementation in Under-Developed Countries: Example of the West African Nations.” in Palvia S., Palvia P. and Zigli R. (eds.), Global issues of information technology management, Idea Group publishing, pp. 187-212, 1992.

Jere I. M., “A new approach to IT in Zambia: Analysis and proposals” in Bhatnagar S.C. (ed.), Information Technology Manpower: Key Issues for Developing Countries, Tata McGraw-Hill Publishing Company Limited, New Delhi, pp. 79-84, 1992.

Jirotka M. and Goguen J. A. (eds.), “Requirements Engineering social and technical issues,” Academic Press Limited, London, 1994.

Keen P. G. W., “Information systems and organisational change,” Communication of the ACM, Vol. 24, No. 1, pp. 24-33, 1981.

Kelle U. (ed.), “Computer-Aided Qualitative Data Analysis: Theory, Methods and Practise,” Sage Publications Ltd, London, 1995.

Kitchenham B., “DESMET: A method for evaluating Software Engineering methods and tools,” Technical Report TR96-09, Department of Computer Science, University of Keele, 1996.

Kitchenham B., “Evaluating software engineering methods and tool,” ACM SIGSOFT software engineering Notes, Vol. 23, No. 5, pp. 21-24, September 1998.

Kitchenham B., Linkman S. and Law D., “DESMET: a methodology for evaluating software engineering methods and tools,” Computing & Control Engineering Journal, Vol.8, No.3, pp. 120-126, June 1997.

Kling R., “Behind the terminal: the critic role of computing infrastructure in effective information systems development and use,” in Cotterman W. and Senn J. (eds.), Challenges and strategies for research in systems development, John Wiley, London, pp. 153-201, 1996.

Klopping I. M. and Bolgiano C. F., “Effective Evaluation of Off-The-Shelf Microcomputer Software,” Office Systems Research Journal, Vol. 9, pp. 46-50, 1990.

Kompass Register, “Kompass register: UK register of British industry and commerce,” Vol. 1-4, Croydon, 1998.

Kontio J., “A Case Study in Applying a Systematic Method for COTS Selection,” Proceedings of the 18th International Conference on Software Engineering (ICSE), pp. 201-209, IEEE Computer Society, 1996.

Kontio J., OTSO: “A Systematic Process for Reusable Software Component Selection,” Univerity of Maryland Technical Report No. CS-TR-3478, 1995.

Laudon K. C. and Laudon J. P., "Essentials of Management Information Systems," Prentice Hall International, Inc., New Jersey, 1995.

Le Quesne P. N., "Individual and Organisational factors and the Design of IPSEs," Computer Journal, Vol. 31, No. 5, pp. 391-397, 1988.

Lee A. S., "A Scientific Methodology for MIS Case Studies," MIS Quarterly, Vol. 13, No. 1, pp. 33-50, 1989.

Liebenau J. and Backhouse J., "Understanding information: An Introduction," MacMillan Press Limited, London, 1990.

Lindgaard G., "Some important factors for successful technology transfer" in Levine L. (ed.), Diffusion, Transfer and Implementation of Information Technology (A-45), Elsevier Science B.V. (North-Holland), pp. 53-65, 1994.

Lindqvist U. and Jonsson E., "A Map of Security Risks Associated with Using COTS," Vol. 31, No. 6, pp. 60-66, June 1998.

Lynex A. and Layzell P. J., "Understanding Resistance to Software Reuse," Proceedings of the 8th International Workshop on Software Technology and Engineering Practice (STEP97), IEEE computer society, Los Alamitos, California, pp. 339-349, 1997.

Lyytinen K. and Hirschheim R., "Information Systems Failures – a Survey and Classification of the Empirical Literature," Oxford Surveys in Information Technology 4, pp. 257-309, 1987.

Maguire D. J, Goodchild M. and Rhind D. W. (eds.), "Geographic Information Systems: principles and applications," Longman Group UK Limited, 1991.

Maiden N. A. and Ncube C., "Acquiring COTS Software Selection Requirements," IEEE Software, Vol. 15, No. 2, pp. 46-56, March/April 1998.

Maiden N.A. and Rugg G., "ACRE: selecting methods for requirements acquisition," Software Engineering Journal, Vol.11, No.3, pp. 183-192, May 1996.

McDermid J. A., "COTS: The Expensive Solution?" in IEE colloquium on COTS and Safety Critical Systems, Digest No: 97/013, IEE Computing and Control Division, York, 1997.

16.1 McDermid J. A., "The Cost of COTS", IEEE Computer, Vol. 31, No. 6, pp. 46-52, 1998

Microsoft Corporation, "The Component Object Model Specification," http://www.microsoft.com/com/resources/comdocs.asp, April 1999.

Miles M. B. and Huberman A. M., "An expanded sourcebook: Qualitative Data Analysis," Sage Publications, London, 1994.

Mohan L., Belardo S. and Bjorn-Andersen N., "A contigency approach to managing information technology in developing countries: Benefiting from lessons learned in developed nations," in Bhatnagar S.C. and Bjorn-Andersen N. (eds.), Information

Technology in Developing Countries, Elsevier Science Publishers, North-Holland, pp. 15-22, 1990.

Mollaghasemi M. and Pet-Edwards J., “Technical briefing: making multiple-objective decisions”, IEEE computer society press, Los Alamitos, California, 1997.

Montealegre R. and Applegate L., “Information Technology and Organisation Change: Lessons from a Less-Developed Country” in Levine L. (ed.), Diffusion, Transfer and Implementation of Information Technology (A-45), Elsevier Science B.V. (North-Holland), pp. 99-121, 1994.

Morisio M., and Tsoukiàs A., “IusWare: a methodology for the evaluation and selection of software products,” IEEE Proceedings of Software Engineering, Vol. 144, No. 3, pp. 162-174, June 1997.

Mumford E., “Designing Human Systems,” Manchester Business School, Manchester, 1990.

Mumford E., “Effective Systems Design and Requirements Analysis: The ETHICS Approach,” Macmillan Press Ltd, Hampshire, 1995.

Mursu A., Olufokunbi K., Soriyan H. A. and Korpela M., “Information systems development in a developing country: theoretical analysis of special requirements in Nigeria and Africa,” Proceeding of the 33rd Hawaii International Conference on System Sciences, IEEE computer society, Los Alamitos, California, 2000.

Nachmias F. and Nachmias D., “Research Methods in the Social Sciences, 5th Edition,” Arnold a member of the Hodder Headline Group, London, 1996.

Nafziger E. W., “The economics of developing countries - 2nd ed.,” Englewood Cliffs, New Jersey, Prentice-Hall, 1990.

Ncube C. and Maiden N.A.M., “Guiding Parallel Requirements Acquisition and COTS Software Selection,” Proceedings of 4th International Symposium on Requirements Engineering, Limerick, Ireland, June 1999.

Nuseibeh B. and Easterbrook S., “Requirements engineering: a roadmap,” in Finkelstein A. (ed.), The future of Software engineering, ACM press, pp. 37-46, 2000.

O’Brien J. A., “Management Information Systems: Managing Information Technology in Internetworked Enterprise, fourth edition,” Irwin McGraw-Hill, Boston, 1999.

Oberndorf P. A., “Facilitating Component-Based software Engineering: COTS and Open Systems,” in Proceedings of the Fifth International Symposium on Assessment of Software Tools, IEEE Computer Society, Los Alamitos, California, pp. 143-148, June 1997.

Oberndorf P. A., Brownsword L. and Morris E., “Workshop on COTS-Based Systems,” Software Engineering Institute, Carnegie Mellon University, Special Report CMU/SEI-97-SR-019, November 1997.

Object Management Group (OMG), "The Common Object Request Broker Architecture (CORBA)," Object Management Group, Framingham, MA, http://www.omg.org/Corba/ Corbaiiop.htm, February 1998.

Odedra M. "IT Applications in the Commonwealth Developing Countries," in Harindranath G. and Liebenau J. (eds.), "Information Technology policies and applications in the commonwealth developing countries," Management and Training Services Division, Commonwealth Secretariat, London, pp. 39-121, 1993a.

Odedra M. "IT Policies in the Commonwealth Developing Countries," in Harindranath G. and Liebenau J. (eds.), "Information Technology policies and applications in the commonwealth developing countries," Management and Training Services Division, Commonwealth Secretariat, London, pp. 9-35, 1993b.

Odedra M., "The role of International Organisations in IT transfer: The African Experience," in Heeks R., Bhatt P., Huq M., Lewis C., Shibli A. (eds.), Technology and developing countries: Practical applications, theoretical issues, Frank Cass & Co. Ltd, London, pp. 215-224, 1995.

Ojo S. O., "Socio-cultural and organisational issues in IT application in Nigeria," in Bhatnagar S. C. and Odedra M. (eds.), Social Implications of Computers in Developing Countries, Tata MacGraw-Hill Publishing Company Limited, New Delhi, pp. 99-109, 1992.

Okot-uma R. W'O, "A perspective of contextual, operational and strategy: Problems of Infomediation in Developing Countries," in Bhatnagar S. C. and Odedra M. (eds.), Social Implications of Computers in Developing Countries, Tata MacGraw-Hill Publishing Company Limited, New Delhi, pp. 10-25, 1992.

Oz E, "When professional standards are lax: the CONFIRM failure and its lessons," Communication of the ACM, Vol. 37, No. 10, pp. 29-36, 1994.

Palvia P., "Global information technology research: past, present and future," Journal of Global Information Technology Management, Vol. 1, No. 2, pp. 3-14, 1998.

Pandit N. R., "The Creation of Theory: A Recent Application of the Grounded Theory Method," The Qualitative Report, Vol. 2, No. 4, 1996.

Pare G. and Elam J. J., "Using case study research to build theories of IT implementation," in Lee A. S., Liebenau J. and DeGross J. I. (eds.), Information Systems and qualitative research, Chapman and Hall, London, pp. 543-568, 1997.

Pasmore W. A and Sherwood J. J. (eds.), Sociotechnical systems: A sourcebook, La Jolla, University Associates Inc., 1978.

Pava C., "Managing New Office Technology: An Organisational Strategy," The Free Press, 1983.

Petrazzini B. and Kibati M., "The Internet in developing countries," Communication of ACM, Vol.42, No. 6, pp. 31-36, 1999.

Pfleeger S.L., "Understanding and improving technology transfer in software engineering," Journal of Systems and Software, Vol 47, pp. 111-124, 1999.

Pohl K., "Requirements Engineering: An Overview," Encyclopia of computer Science and Technology, Marcel Dekker, Inc, New York, Vol. 36, No.21, pp. 345-386, 1997.

Poltrock S. E. and Grudin J., "Organisational obstacles to interface design and development: two participant-observer studies," ACM Transaction on Computer-Human Interaction, Vol. 1, No. 1, pp. 52-80, 1994.

Potts C., "Software engineering research revisited," IEEE Software, September 1993, pp. 19-29, 1993.

Powell A., Vickers A., Lam W. and Williams E., "Evaluating Tools to support Component Based Software Engineering," in Proceedings of the Fifth International Symposium on Assessment of Software Tools, IEEE Computer Society, Los Alamitos, California, pp. 80-89, June 1997.

Pressman R. S., "Software Engineering: A practitioner's approach – fifth edition," McGraw-Hill, New York, 2000.

Prevost Y. A. and Gilruth P., "Environmental Information Systems in Sub-Saharan Africa," Technical Paper No. 12, Africa Technical Department, World Bank, Washington, 1997.

Puma Systems, Inc., "Commercial-Off-The-Shelf System Evaluation Technique (COSSET)," http://www.pumasys.com/cosset.htm, March 1999.

Rader J., "Mechanisms for Integration and Enhancement of Software Components," in Proceedings of the Fifth International Symposium on Assessment of Software Tools, IEEE Computer Society, Los Alamitos, California, pp. 24-31, June 1997.

Robey D., Gupta K. S. and Rodriguez-Diaz A., "Implementing Information Systems in Developing Countries: Organisational and cultural considerations," in Bhatnagar S.C. and Bjorn-Andersen N. (eds.), Information Technology in Developing Countries, Elsevier Science Publishers, North-Holland, pp. 41-50, 1990.

Robson C., "Real World Research: a resource for social scientists and practitioner-researchers," Blackwell Publishers Ltd, Oxford, 1993.

Rowley J. E. "Selection and Evaluation of Software," Aslib Proceedings, Vol. 45, pp. 77-81, 1993.

Roy A. N., "The Third World in the age of globalisation: requiem or new agenda?" Madhyam Books, Delhi, 1999.

Roy, B., "The Outranking Approach and the Foundations of ELECTRE Methods," Theory and Decision, Vol. 31, pp. 49-73, Kluwer Academic Publishers, Netherlands, 1991.

Saaty T. L., "The Analytic Hierarchy Process," McGraw-Hill, New York, 1990.

Sapsford R., “Survey Research,” Sage Publications, London, 1999.

Sauer C., “Why Information Systems Fail: A Case Study Approach,” Alfred Waller Ltd, Oxfordshire, 1993.

Sawyer S., “Packaged software: implications of the differences from custom approaches to software development,” European Journal of Information Systems, Vol. 9, pp. 47-58, 2000.

SEL, “SEL Package-Based System Development Process,” Software Engineering Laboratory, http://sel.gsfc.nasa.gov/doc-st/tech-st/pbs-dev.html, February 1996.

Sena J. A. and Shani A. B., “Intelligence systems: A sociotechnical systems perspective,” in: SIGCPR’99. Proceedings of the 1999 ACM SIGCPR conference on Computer personnel research, pp. 86-93, 1999.

Sessions R., “Component-oriented middleware for commerce systems,” IEEE Software, Vol. 15, No. 5, pp. 42-43, 1998.

Shitima M. N., “Information technology in Zambia,” in Bhatnagar S.C. and Bjorn-Andersen N. (eds.), Information Technology in Developing Countries, Elsevier Science Publishers, North-Holland, pp. 87-100, 1990.

Sledge C. and Carney D., “Case Study: Evaluating COTS Products for DoD Information Systems,” SEI Monographs, http:www.sei.cmu.edu/cbs/ monographs.html, July 1998.

Sommerville I. and Rodden T., “Human, Social and Organisational Influences on the Software Process, Technical Report CSEG/2/1995,” Computing Department, Lancaster University, 1995.

Sommerville I. and Sawyer P., “Requirements Engineering: A good practice guide,” John Wiley and Sons, Chichester, 1997.

Sommerville I., “Software Engineering,” Addison Wesley Longman Limited, Essex, 1995.

Specter C. N. and Sahay S., “Information Technology for Natural Resource Inventory and Monitoring in Developing Countries,” in Palvia S, Palvia P. and Zigli R. (eds.), Global issues of information technology management, Idea Group publishing, 1992, pp. 146-170.

Spectrum, “Moving into the Information Age: An International benchmarking study,” Department of Trade and Industry, http:www.isi.gov.uk/isi/bench/mitia/index, 1998.

Stamelos I., Vlahavas I., Refanidis I. and Tsoukias A., “Knowledge based evaluation of software systems: a case study,” Information and Software Technology, Vol. 42, pp. 333-345, 2000.

Stavridou V., “Integration in software intensive systems,” Journal of Systems and Software, Vol. 48, pp. 91-104, 1999.

Strauss A. and Corbin J., "Basics of qualitative research: Grounded theory procedures and techniques," Sage Publications limited, London, 1990.

Sun Microsystems Inc., "Enterprise JavaBeans Specification 1.0," http://java.sun.com /products/ejb/docs.html, March 1998.

Szyperski C., "Component Software: Beyond Object-Oriented Programming," Addison Wesley Longman Limited, Essex, 1998.

Tallon P. P. and Kraemer K. L., "The impact of technology on Ireland's economic growth and development: Lessons for developing countries," Proceeding of the 32nd Hawaii International Conference on System Sciences, IEEE computer society, Los Alamitos, California, 1999.

Tan B. C. Y., Wei K. and Watson R. T., "The equalizing impact of a group support system on status differentials," ACM Transaction on Information Systems, Vol. 17, No. 1, pp. 77-100, 1999.

Taylor J. C. and Felten D. F, "Performance by Design: Sociotechnical Systems in North America," Prentice Hall, Englewood Cliffs, 1993.

Todaro M. P., "Economic Development", 5th ed., Longman Publishing, New York, 1997.

Tran, V., Liu Dar-Biau and Hummel B., "Component-based systems development: challenges and lessons learned," Proceedings of the Eighth IEEE International Workshop on Software Technology and Engineering Practice incorporating Computer Aided Software Engineering, IEEE Computer Society, Los Alamitos, California, pp. 452-462, 1997.

Trist E. L. "The Sociotechnical Perspective," in Van de Ven A. H. and Joyce W. E. (eds.), Perspective on Organisation Design and Behaviour, John Wiley, New York, pp. 22-23, 1981.

Trist E.L, "On Socio-Technical Systems" in Pasmore W. A and Sherwood J. J. (eds.), sociotechnical systems: A sourcebook, La Jolla, University Associates Inc., 1978.

Trist E.L. and Bamforth K.W. "Some social and psychological consequences of the Longwall method of coal-getting," Human Relations, Vol. 4, pp. 3-38, 1951.

Turban E., McLean E. and Wetherbe J., "Information technology for management: improving quality and productivity," John Wiley, New York, 1996.

UKAIS, "The UKAIS Newsletter," Vol. 5, No. 3, November 1999.

UNDP, "Zambia Human Development Report 1998" United Nations Development Programme, 1998.

USAID, "Zambia in Brief: Economy," http://www.usaid.gov/zm/facts.html, 2000.

Vidgen R., “Stakeholders, soft systems and technology: separation and mediation in the analysis of information system requirements,” Information Systems Journal, Vol. 7, pp. 21-46, 1997.

Vidgen R., “Using the Multiview2 framework for internet-based information systems development,” in Wood-Harper A. T. Jayaratna N. and Wood J. R. G. (eds.), Methodologies for Developing and Managing Emerging Technology based Information Systems, pp. 389-403, Springer-Verlag London Limited, 1999.

Vigder M. R. and Gentleman W. M. and Dean J., “COTS Software Integration: State of the art,” National Research Council, Canada, NRC Report Number 39198, January 1996.

Vincke P., “Multicriteria decision-aid,” Wiley publishing, Chichester, 1992.

Voas J. M. and McGraw G., “Software Fault Injection: Inoculating Programs Against Errors,” John Wiley & Sons, New York, 1998.

Voas J. M., “Disposable information systems: the future of software maintenance,” Journal of Software Maintenance: Research and Practice, Vol. 11, pp. 143-150, 1999.

Vozikis G. S., Goss E. and Mescon T. S., “Technical factors affecting International Information and Technology Transfer,” in Palvia S, Palvia P. and Zigli R. (eds.), Global issues of information technology management, Idea Group publishing, pp. 450-462, 1992.

Walsham G., Symons V. and Waema T., “Information Systems as social systems: Implications for developing countries,” in Bhatnagar S.C. and Bjorn-Andersen N. (eds.), Information Technology in Developing Countries, Elsevier Science Publishers, North-Holland, pp. 51-61, 1990.

Walters N., “Systems Architecture and COTS Integration,” Proceedings of SEI/MCC Symposium on the use of COTS in systems Integration, Software Engineering Institute Special Report CMU/SEI-95-SR-007, June 1995.

Weisberg H. F., Krosnick J. A. and Bowen B. D., “An Introduction to Survey Research, Polling and Data Analysis,” Sage Publications, London, 1996.

Weyuker E. J., “Evaluation techniques for improving the quality of very large software systems in a cost-effective way,” Journal of Systems and Software, Vol. 47, pp. 97-103, 1999.

Wieringa R. J., “Requirements Engineering: Frameworks for understanding,” John Wiley and Sons, Chichester, 1996.

Williams F., “Appraisal and evaluation of software products,” Journal of Information Science, Principles and Practice, Vol. 18, pp. 121-125, 1992.

Wilson S., Bekker M., Johnson P. and Johnson H., “Helping and hindering user involvement- a tale of everyday design,” in CHI’97 Conference proceedings on Human factors in computing systems, 1997.

Woherem E. E., “IT Manpower Development Strategy at the Organisational level,” in Bhatnagar S.C. (ed.), Information Technology Manpower: Key Issues for Developing Countries, Tata McGraw-Hill Publishing Company Limited, New Delhi, pp. 147-157, 1992a.

Woherem E. E., “Strategy for Indigenisation of IT in Africa,” in Bhatnagar S. C. and Odedra M. (eds.), Social Implications of Computers in Developing Countries, Tata MacGraw-Hill Publishing Company Limited, New Delhi, pp. 70-80, 1992b.

Wood J. and Silver J., “Joint Application Development,” John Wiley & Sons Inc, New York, 1995.

World Bank, “Country information centre: Zambia country profile,” http://www.ifc.org/abn/cic/zambia/english/prof.htm, 1998.

World Bank, “The World Development Report 1999/2000,” Oxford University Press, 2000.

Yin R. K., “Case Study Research: Design and Methods,” Sage Publications, London, 1994.

Yoon, K. and Hwang C., “Multiple Attribute Decision-Making: an Introduction,” Sage Publisher, 1995.

Zave P., “Classification of Research Efforts in Requirements Engineering,” ACM Computing Surveys, Vol. 29, No. 4, pp. 315-321, 1997.

Zviran M., “A Comprehensive Methodology for Computer Family Selection,” Journal of Systems Software, Vol. 22, pp17-26, 1993.