Hamdan, Khaled (2009) An Investigation into Software Estimation Methods. Doctoral thesis, University of Sunderland.

Downloaded from: http://sure.sunderland.ac.uk/id/eprint/3557/

Usage guidelines

Please refer to the usage guidelines at http://sure.sunderland.ac.uk/policies.html or alternatively contact sure@sunderland.ac.uk.

AN INVESTIGATION INTO SOFTWARE ESTIMATION METHODS

KHALED H. M. HAMDAN

IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

SCHOOL OF COMPUTING & TECHNOLOGY

UNIVERSITY OF SUNDERLAND, UNITED KINGDOM

FEBRUARY, 2009

ACKNOWLEDGEMENTS

I wish to express my deepest gratitude and admiration to all those who made this thesis possible. I would like to thank my advisor Professor Peter Smith for his significant influence on my research and for his continuous support and encouragement. His guidance as well as devotion of his time and effort helped to develop this work into its present form. Without this, the thesis would have been a more difficult mission.

I thank Dr. J. Moses, Dr. F. abu Sitta, Dr. H. El-Khatib, Professor B. Belkhouche, Dr. T. Zoubeidi, Professor M. Madi and Dr. L. Angelis for their deep interest in my research work and for the help they come up with, many good ideas and suggestions about my research, valuable advice and very good perception.

A huge debt of gratitude goes to my wife and family. I dedicate this work to my beloved wife and children who have provided me with the inspiration, without that, I could not pursue my dream. I am also grateful to my parents and my brothers and sisters for their support, love and encouragement.

I thank my friends and colleagues in UAE University. I am especially indebted to the management of UGRU for their support. Finally, I wish to thank the IT organisations in UAE that supplied the data which made this study possible. I thank you all.

Khaled Hamdan

DECLARATION

I hereby declare that the work presented in this theses was carried out by myself at Sunderland University, except where due acknowledgment is made, and has not been submitted for any other degree.

Khaled Hamdan

February 2009

To my Parents and family

Table of Contents

List of Figures

List of Tables

ABSTRACT

There are currently no fully validated estimation approaches that can accurately predict the effort needed for developing a software system (Kitchenham, et al, 1995). Information gathered at the early stages of system development is not enough to provide precise effort estimates, even though similar software systems may have been developed in the past. Where similar systems have been developed, there are often inherent differences in the features of these systems and in the development process used. These differences are often sufficient to significantly reduce estimation accuracy. Historically, cost estimation focuses on project effort and duration. There are many estimation techniques, but none is consistently 'best' (Shepperd, 2003).

Software project management has become a crucial field of research due to the increasing role of software in today's world. Improving the functions of project management is a main concern in software development organisation. The purpose of this thesis is to develop a new model which incorporates cultural and leadership factors in the cost estimation model, and is based on Case-Based Reasoning. The thesis defines a new knowledge representation "ontology" to provide a common understanding of project parameters. The associated system uses a statistically simulated bootstrap method, which helps in tuning the analogy approach before application to real projects. This research also introduces a new application of Profile Theory, which takes a formal approach to the measurement of leadership capabilities.

A pilot study was performed in order to understand the approaches used for cost estimation in the Gulf region. Based on this initial study, a questionnaire was further refined and tested. Consequently, further surveys were conducted in the United Arab Emirates. It was noticed that most of the software development projects failed in terms of cost estimate. This was due to the lack of a precise software estimation model. These studies also highlighted the importance of leadership and culture in software cost estimation.

Effort was estimated using regression and analogy. The Bootstrap method was used to refine the estimate of effort based on analogy, with correction for bias. Due to the very different nature of the core and support systems, a separate model was developed for each of them. As a result of the study, a new model for identifying and analysing was developed. The model was then evaluated, and conclusions were drawn. These show the importance of the model and the factors of organisational culture and leadership in software project development and in cost estimation. Potential areas for future research were identified.

CHAPTER 1

Introduction

1.1 Motivation

Software project effort estimation is an increasingly important subject, due to the overwhelming role of software in today's global market (Kitchenham, et al, 1995). However, there is currently no optimal approach to accurately predict the effort needed to develop a software system (Shepperd and Schofield, 1997). Information gathered at the early stages of software system development is often insufficient to provide precise effort predictions. Even if data exist from previous projects, there will commonly be new facets of development in any new project. This means that it is difficult to produce an accurate estimate of system development effort at an early stage in the project. There are many estimation techniques, but none is consistently 'best' (MacDonell and Shepperd, 2003).

1.1.1 The importance of software project effort estimation

Many organisations invest greatly in software development, and are losing large parts of their revenue to troubled software projects (Masticola, 2007). The Standish Group[1] "CHAOS Report" reported that, in the year 2004, only 29% of software projects in large enterprises succeeded (i.e. produced acceptable results that were delivered close to 'on-time' and 'on-budget'). Of the remainder, 53% were 'challenged' (i.e. significantly over budget and schedule) and 18% failed to deliver (Johnson, 2006).

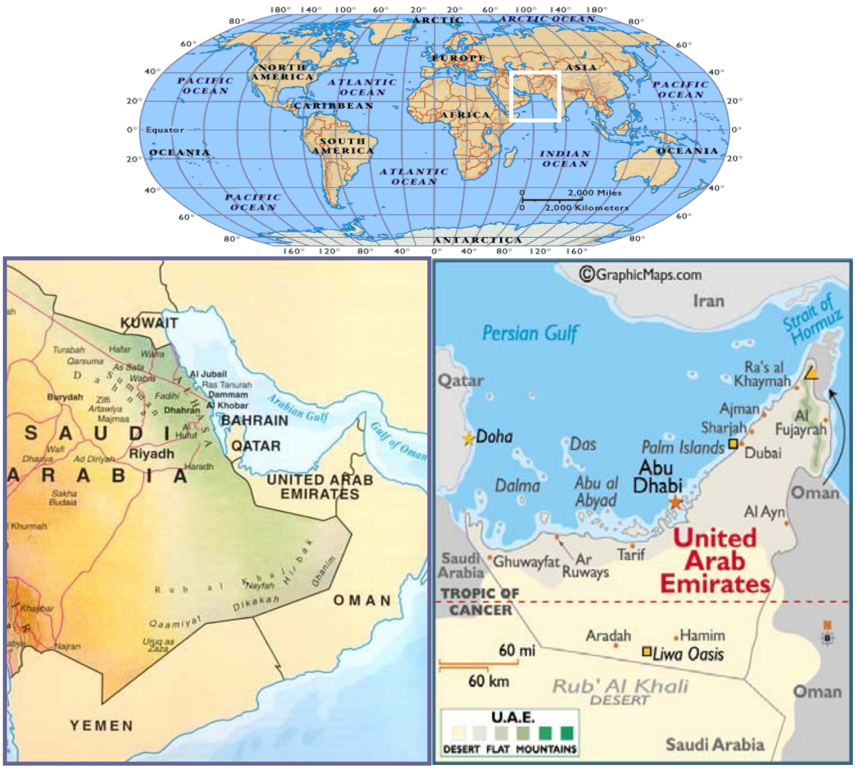
[1] See http://www.standishgroup.com

Organisations and government departments spend approximately one trillion dollars on Information Technology (IT) hardware, software and services worldwide (Charette, 2005). In the year 2005, 5 to 15 percent of IT projects were abandoned before or shortly after delivery. Many others missed deadlines or were over budget and required massive reworking (Charette, 2005).

According to R. N. Charette (2005) in "Why Software Fails", in 2003, the United Kingdom had more than 100 major governmental IT projects, totalling 20.3 billion dollars. In 2004, the U.S. government had 1200 civilian IT projects costing more than 60 billion dollars, and 16 billion dollars was spent on military software. According to the David Consulting Group, a U.S. government study on software development projects revealed that 60% of projects were behind schedule, 50% were over budget, and 45% of delivered projects were unusable (Garmus, 2006).

1.1.2 The Arab Gulf States

The Gulf States consist of Bahrain, Kuwait, Oman, Qatar, Saudi Arabia and the United Arab Emirates (UAE) (see Figure 1). The data was selected solely from UAE but given the similarities between UAE and Gulf States, the results of this study were expected to apply to Gulf States (Hamdan et al., 2005). These all have similar cultural environments, political systems, leadership approaches and economic strengths. The Gulf States have particular cultural aspects which may impact on development projects (Hofstede, 1991). Within the Gulf States, the UAE is one of the fastest growing countries in utilising, adopting and developing IT. It is one of the Gulf Co-operation Council (GCC) states. Local and federal government agencies have spent millions of dollars on internal software development. However, a big proportion of the developed applications either go over budget or are not delivered on time (Hamdan et al, 2005).



(a) (b)

Figure 1.1: (a) The geography of the Arab Gulf States (b) The UAE
Source: http://www.gulf.destinations.starwoodhotels.com and GraphicMap.com

The UAE is a young nation categorised by rapid and ongoing development across all sectors of its society (Harold, 2006), in which the discovery of oil has created multi-economic opportunities for a mixed workforce drawn from all over the world (Yousef, 1998). The UAE wishes to maintain its leadership among Arab countries in the IT and knowledge economy areas (UNstats[2], 2005; the Madar Research Group[3], 2006). The UAE made the top 50 countries in the United Nation's (UN's) Global e-government report (2005), ranking 42nd place

[2] See http://unstats.un.org/unsd/demographic/products/indwm/wwpub.htm

[3]See http://www.MadarResearch.com

among the UN's 191 member countries. According to the Madar Research Group (2006), the UAE IT market in the year 2005 was worth approximately US $1.53 billion. This includes software products, IT services and data communications equipment.

According to the UN's Global (2005) and Middle East Times[4] (Claude, 2006), the UAE in general, and Dubai in particular, is the business centre of the Middle East and records the highest Gross Domestic Product (GDP) in the world, at 133.8 billion US dollars. The emirate of Abu Dhabi observed the strongest growth in the country with a 61 percent contribution to GDP in 2005, while the equivalent contribution for Dubai was 27 percent. The UAE per capita GDP is estimated to be 29,751 US dollars. Moreover, the average per capita GDP for the Gulf Cooperation Council (GCC) countries for 2005 was 13,500 US dollars.

1.1.3 Leadership

The success of a software project requires the work of highly capable and motivated individuals; but even the best individuals need someone to lead them (Stellman and Greene, 2006). Researchers have repeatedly demonstrated that lack of leadership in a project is often a cause of that project's ultimate failure (Brown, 1998). The leader of an organisation has an essential role to play in setting the vision and culture for the organisation, and this will impact upon development projects.

[4] See http://www.metimes.com/storyview.php

In the Gulf States, leadership tends to be more hierarchical than in the Western world where it often takes a more functional role (Schneider and Barsoux, 2003). In the Gulf States, leadership intimately involves personality and is distinguished from Western leadership by the influence of Arab authority values (Neal et al., 2005) which are used to maintain explicit leadership. This has been expressed more generally by Abdulla and Al-Homoud (Neal et. al, 2005).

Abdulla and Al-Homoud (2001) applied the GLOBE (Global Leadership and Organisational Behaviour Effectiveness) instrument to evaluate leadership in the Gulf States. They found great similarities between the implicit leadership models applied across the Gulf region as a whole. They found that the most outstanding and positive leadership behaviours were administrative competence, diplomacy, vision, integrity, performance-orientation and inspiration. On the other hand, the leadership traits which they believed to inhibit success were non-participation, autocratic behaviour, autonomous decision-making, malevolence, face-saving, and self-centred styles (Abdulla and Al-Homoud, 2001).

In the study of the parameters that impact upon software effort and cost estimation in government departments, it is proposed here that leadership characteristics are important and should be incorporated within a cost estimation model.

1.1.4 Analogy and Case-Based Reasoning (CBR)

This research seeks to improve estimation accuracy by building on existing analogy methods, such as those of Shepperd and Schofield (1997). Within analogy, it is proposed to incorporate the features of organisational culture and leadership. This study will be the first to demonstrate the impact and the influence of these attributes. These and other identified attributes will then be used in a system for effort estimation (Hamdan et al., 2006).

Analogy is an approach used to improve effort estimation by understanding and measuring the similarity between cases (Shepperd, et al, 1996; Angelis and Stamelos, 2000). Analogy uses similar, completed, projects (source cases) as the basis of the estimate for new projects (target cases). Analogy methods are thus based on data from actual projects, avoiding the reliance on recall (Shepperd and Schofield, 1997).

Case-based reasoning (CBR) is a problem solving paradigm that does not require identical problems that have been previously solved, but tries to predict an outcome by finding similar cases to the current problem (Aamodt and Plaza, 1994; Shepperd, 2003). The major strategy of CBR is saving previous experiences into a case base in order to propose solutions to new problems (cases). A new problem is solved by finding a similar past case, and reusing the solution in the new problem situation.

CBR is accepted as an alternative to traditional numerical models (Shepperd and Schofield, 1997). Cases may exactly or partly repeat themselves over time. The more experience in estimation and development that exists in a specific area, the easier developers will find it to produce estimates, since they have built up knowledge about previous cases (Shepperd and Schofield, 1997). Some cases may not repeat themselves exactly but may share certain similarities.

Estimation by analogy is a form of CBR which can be easily applied to any type of variables, and can be interpreted in a straightforward manner (Boehm, 1981). The terms analogy and CBR are often used interchangeably although the former emphasises the cognitive and the latter the computer science (Shepperd, 2003). A database of past projects is used as a reference point in order to combine actual costs of previous projects for the prediction of the costs of a new project with similar attributes. CBR can be applied either at the project level as a whole, or at the sub-system level. Project attributes include line of business, programming language, personnel experience, functionality and cultural aspects. Attributes may be categorised quantitatively (e.g. functionality can be measured by counting the number of function points and feature points) or more qualitatively (e.g. team capability can be measured by ordinal scale values such as ‘‘low’’, ‘‘nominal’’ and ‘‘high’’).

Various research papers (Shepperd, et al, 1996; Angelis and Stamelos, 2000; Lederer and Prasad, 1998; Shepperd and Schofield, 1997) discuss different approaches to CBR which improve cost estimation by using knowledge from past cases. However, CBR estimation accuracy can still be improved upon (Shepperd and Schofield, 1997) and for this reason there is usually the need to combine more than one estimation technique.

Analogy and CBR have many merits, including simplicity and interpretability. Moreover, analogy has a wide range of application since it requires virtually no theoretical assumptions. To apply, it needs reliable historical data in order to select similar cases. However, there are several disadvantages of using analogy:

- Similar projects may not exist,
- Historical data may not be accurate,
- Case data can be hard to gather,
- It is necessary to build a library of cases before the system can be useful,
- Cases require interpretation; two projects that may seem similar may indeed be different in a critical way,
- The uncertainty in assessing similarity and difference means that two different analysts may have significantly different views and thus result in very different estimates.

1.2 Research Questions

The primary aim of this research is to develop an improved model for software project cost estimation.

Towards this end the research will focus on the following questions:

- How important are the factors of organisational culture and leadership in software project development?
- Can a cost estimation model be devised, that takes leadership and organisational culture into account?
- Are there software development problems which are particular to the Gulf States? (The model is tested on data from the Gulf States; however, it is believed that the model could be applicable elsewhere).
- Can Case-Based Reasoning (CBR) be used to improve effort estimation in software development projects?

The first two points in the research questions have been raised in the literature. For instance, Gardner (1990), Sarros et al. (2006), and Fairholm (1991), all noted the value of leadership and individual awareness on the culture of the organisation. An organisation will manage more productively when its values are shared by its teams. Furthermore, organisational culture incorporates a set of assumptions, beliefs, and values which guide the function of members of the organisation (Hofstede and Hofstede, 2005; Schein, 2004). These assumptions typically include personality traits, power relationships, behaviours and values.

Concerning the fourth and final point, the fact that CBR is derived from human reasoning gives the approach a significant advantage in software development cost estimation. This work will build upon previous work by (Shepperd and Schofield, 1997). In particular; the work will be the first to incorporate organisational, cultural and leadership aspects to improve effort estimation. It is envisaged that adding these new attributes will improve the accuracy of the model. This hypothesis finds support in the literature. For example, (Boehm, 1981; Kemerer, 1997) argue that the quality of a software team (i.e. capability of programmers and analysts) is a significant factor in determining the cost and quality of a software product (Krishnan, 1998). Cases require interpretation, as CBR does not necessarily provide the "correct" solution for a problem; it merely provides suggestions of possible solutions (Kitchenham et al., 1995). Shepperd and Schofield (1997) similarly faced with diverse interpretations of measurement parameters; recommend that an ontology be developed to support a consistent measurement strategy.

The proposed method is based on survey results, a literature survey and identifies the “needs” of the community for a model based on an ontology-based cost estimation process framework for defining the semantics of project development data. This ontology proposes to reduce the risk of misunderstandings by unifying the terminology of the different stakeholders. In this study an ontology is presented to support the estimation of software project effort incorporating cultural factors. These factors are then used in a CBR system for effort estimation. This system enables a project manager to elicit software project factors, features and terms that are semantically equivalent to those used in a previous project.

1.3 Research Methodology

The following research steps were used:

- Evaluation of the limitations of current methods using a literature survey; in particular covering the use of case-based reasoning.
- Identification of problems of current methods using questionnaires and surveys. This took the form of:
    - Questionnaire design
    - Pilot study
    - Structured interviews
    - Survey 1
    - Survey 2
    - Analysis of results and conclusions drawn
- Propose new model based on the survey results, and “needs” of the community.
- Develop a model based on case-based reasoning.
- Analyse and evaluate the model, using data from real projects.

- Run model on past and present case studies.
- Analyse results and draw conclusions.

1.4 Thesis Structure

The thesis consists of eight chapters. This chapter has highlighted the rationale for the work and the research questions. Chapter 2 examines models that have been previously proposed for software cost estimation. In particular their shortcomings are elaborated with respect to the aims of this work. Chapter 2 also provides a review of the importance of leadership and culture in software development and concentrates its attention on traits, behaviours, and crucial environmental factors. Chapter 3 details the surveys undertaken, and presents the questionnaire development and data collection procedure. A statistical analysis of the data presented in Appendix D was collected from the different organisations surveyed. Chapter 4 describes the measurement of attributes, protocol and the different parameters used in the survey. Thus, the necessary requirements for scientific measurement and ontology-based representations are outlined. Chapter 5 presents the new software effort estimation model, which is based on the survey results, literature survey and the needs of the community. The results and evaluation of the model are presented and discussed in Chapter 6. Chapter 7 draws conclusions from the work and presents future areas for study which might further improve software cost estimation methods.

CHAPTER 2

Software Effort Estimation and related issues: a review

2.1 Introduction

Commercial and government organisations spend large amounts of money on IT infrastructure and software development in an effort to provide competitive services or products to their clients. At the beginning of each year, and as part of the planning process, those organisations allocate budgets for their upcoming projects. The existence of models or tools to estimate the software development budget gives organisations a competitive advantage, helps them reduce unforeseen costs, and enables them to provide better services and products.

A lot of research has been undertaken in an effort to accurately determine the cost of software development projects in their initial stages (Albrecht, 1979; Boehm, 1981; Putnam, 1978; Rubin, 1983). However, none of the models proposed have been completely successful in precisely predicting the effort needed for a particular software development project (Kemerer, 1987).

In this chapter, some of the most popular effort estimation models that have been developed over the years will be discussed. The strengths and the weaknesses of the models will be reviewed. The majority of software cost estimates can be mapped into three categories: Algorithmic Estimation Models, Expert Judgement and Case-Based Reasoning. In section 2.3, these models will be discussed in detail. The role of leadership is discussed in section 2.4. The importance of leadership and culture characteristics are also explored here, as are some basic elements of project team leadership, such as personality.

2.2 Current issues

There has been a continuous search for better models and tools to aid project managers in the estimating process (Kitchenham, et al, 1995; Jorgensen and Shepperd, 2007; Futrell, Shafer and Shafer, 2002). Software cost estimation is an essential activity throughout the software life cycle. Cost estimation may be performed before, or during software development and, after the project is complete, the actual values are compared with estimates to determine the accuracy of the estimation. Good estimates are useful for project productivity assessment, for initial validation, for monitoring the project's progress, and for deciding whether the project ought to proceed. Inherent difficulties, such as data availability, the number of parameters being measured, complicate the process. Moreover, there is a lack of a consistent interpretation of project attributes within an organisation and across organisations. Thus, improving project cost estimation is a complex yet necessary endeavour in software development.

2.3 Estimation Methods

The estimation of effort and duration is one of the most critical activities during the software life cycle; a task known as software cost estimation. Commonly software cost estimates are based on the following methods: Algorithmic Estimation Models (COCOMO-Boehm, 1981; SLIM-Putnam, 1978, and Function Points-Albrecht, 1979), Expert Judgement (Hughes, 1996) and Case-Based

Reasoning (Mukhopadhyay et al. 1992; Aamodt & Plaza 1994; Shepperd et al., 1996). All three approaches have obvious advantages and disadvantages.

Algorithmic Estimation Models (AEM) are very useful when used correctly and calibrated with historical data reflecting the characteristics of the project. Algorithmic cost estimation involves the application of a cost model which is usually a mathematical formula derived through statistical data analysis. The main advantages of algorithmic methods are objectivity and capability to produce repeatable results. Their main disadvantage is that they are often built using data from quite old projects and therefore they may not reflect the current environment and development situations.

Expert Judgement relies on the experience of one or more experts. Expert Judgement can be relatively easy to apply and can produce fast evaluation but suffers from the difficulty of finding real experts and can suffer from subjective assessments. The most common way that experts make cost estimates and formulate judgements is by the use of analogies or comparisons (Hughes, 1996; Boehm, 1981).

Case-based reasoning (CBR) is an improved form of estimation by analogy (Shepperd and Schofield, 1997) that compares the software project under consideration with similar historical projects. CBR concentrates on a concrete, well-defined estimation framework and can be used, provided that suitable past projects can be found and that the mechanism for applying analogy is correct.

Each of these approaches are reviewed in detail in the sections which follow:

2.3.1 Algorithmic Estimation Models (AEM)

In their earliest efforts, researchers used algorithmic estimation models as a tool for software development effort estimation. AEM models depend totally on mathematical and numerical calculation, and do not take into consideration analogy with other similar projects. In this section, Line of Code (LOC) and Function Point (FP) models will be discussed.

2.3.1.1 The Nelson Model

In the early 1960s, Nelson and his team developed an effort estimation model. Their model was based on the study of 104 attributes and 169 software projects (Nelson, 1966). This model has formed the foundation for many other models being used today. It was also used and tested by the U.S. Air Force System Development Corporation (Boehm and Sullivan, 2000). However, with the advancement and complexity of software applications and tools, this model has become somewhat outdated. The model has also shown major disadvantages because of inaccurate estimation for non-linear cases.

2.3.1.2 The Walston-Felix Model

Walston and Felix (1977) extended the work of Nelson and incorporated new parameters. Their research was conducted in the early 1970s, and was based on calculating software development productivity, size, and cost. They tried to create cost models for early stages of the life cycle (Boehm, 1981). Their model used estimated and actual source lines for productivity estimates. In addition, program length was used as a tool to predict program characteristics, reliability and ease of maintenance. Walston and Felix performed some of the early work that led to the first generation of software effort estimation techniques using project parameters such as effort, average staff size, as well as total costs. Their work was extended to cover many different situations. It fits a non-linear model but does not work when applied to subsets (Boehm, 1981).

2.3.1.3 The Boehm Model

In 1981, Boehm developed COCOMO (COnstructive COst MOdel) which was derived from the analysis and the observation of 63 software development projects (Boehm, 1981). The most crucial calculation in the COCOMO model is the use of the effort equation to estimate the number of person-months required to develop a project. Most other COCOMO results, including estimates for requirements and maintenance, are derived from this quantity (Boehm, 1981). To model complexity, COCOMO projects were categorised into three modes: organic, semi-detached, and embedded. COCOMO used additional variables called “cost drivers” as part of the project’s attributes (Futrell, Shafer and Shafer, 2002; Sommerville, 1995).

COCOMO II was introduced in 1995 as an extension of the 1981 COCOMO model. The model had three sub-models: Applications Composition, Early Design and Post-Architecture, which can be combined in various ways (Reifer, et al., 1999).

The main advantage of the COCOMO Model is that it is considered to be more transparent than other models. The cost drivers are also useful in understanding their impact on projects. In addition, the model is flexible enough to allow for different modes and levels of complexity. Finally, the COCOMO model works well with projects that are similar in size, process or complexity (Futrell, Shafer and Shafer, 2002; Snell, 1997).

Although the COCOMO model has many advantages, it also has many disadvantages. One of the main disadvantages is that it is difficult to estimate Lines of Code (LOC) early in the project. Secondly, it is dependent on the knowledge of cost drivers and/or the amount of time spent in each phase (Futrell, Shafer and Shafer, 2002). Finally, it is extremely vulnerable to mis-classification of the development mode (Snell, 1997).

2.3.1.4 Putnam’s Software Life-Cycle Model (Putnam’s SLIM)

Putnam’s SLIM model was developed in the late 1970s (Putnam, 1978). It uses an automated mathematical model for software estimation based on Putnam’s analysis of software products in terms of the Rayleigh function (Norden, 1958; Putnam and Myers, 1992). SLIM enables software cost estimators to perform the following functions: calibration (fine tuning the model to represent the software development environment of the historical database of past projects), building (gathering software characteristics, personal attributes, computer attributes for system software) and sizing (using lines of code (LOC) costing technique) (Boehm et al., 2000).

Putnam's model is based on the non-linear Norden Rayleigh manpower distribution and uses the analysis of many completed projects. The central part of Putnam's model is called the software life cycle equation:

$$y = 2K \text{ at } \exp(-at^2)$$

Where y is the manpower required in time period t, K = total project effort in staff-year, and (a) is the shape parameter, depending upon the point in time at which y reaches its maximum $a = 1 \div 2t_d^2$, and t_d = development time. Putnam argued that the total of the individual cycle curves results in a Rayleigh Curve (Norden, 1958) because software development is implemented in a functionally similar way. The Rayleigh curve used to define the distribution of effort is modelled by a differential equation.

Some of the advantages of the SLIM model are that it depends on a thorough set of software development management tools which sustain the software program's entire life cycle. It also uses linear programming, program evaluation, statistical simulation, and review techniques to derive software cost and effort (Putnam 1978). It thus generates an estimate from few parameters and gives useful guidelines in project management. However, some of the disadvantages of the SLIM are that it works well with large projects but not for small software projects. Finally, SLIM's estimates are considered to be extremely sensitive to technology factors (Snell, 1997).

2.3.1.5 The Jensen Model

The Jensen model is considered similar to Putnam's SLIM model (Jensen, 1984). However, the main advantage of the Jensen model was that it eliminated some of the negative behaviours of Putnam's SLIM model. Both Jenson and Putnam apply the constraint that effort divided by the cube of the development time is less than some constant (which is chosen based on product and project parameters) (Jensen, 1984). One disadvantage of Jensen's work compared to

Putnam's is that it is less sensitive to schedule compression. In addition, the model is applicable to smaller and medium sized projects.

SEER-SEM (System Evaluation and Estimation of Resources) is a proprietary model developed by Galorath, Inc. of Segundo, California. The model covers every aspect of the project life-cycle (Jensen, 1984; Boehm et al., 2000; NASA JSC, 2002). One advantage of SEER-SEM is that it can be applied to all types of software projects. The model also allows and offers tools that address not only software, but also hardware issues (Boehm et al., 2000; NASA JSC, 2002). The disadvantage of this approach is difficult to estimate early in stage and many costs not considered when estimate.

2.3.1.6 The Bailey–Basili META Model

The Bailey-Basili meta-model (Bailey and Basili, 1981) presented a statistical generation process for developing a local resource estimation model. The statistical generation process consists of three steps:

- Computing background equation,
- Determining factors explaining the differences between actual project data and the mean of the estimate derived by the background equation,
- Using a model to predict new project effort.

Bailey and Basili identified about 100 attributes as likely contributors to the variance in the predicted effort, grouping those attributes in a way that shows the positive and negative impact of each on effort. The logical groups that were derived were (Boehm et al., 2000):

- Total Methodology,
- Cumulative Complexity,
- Cumulative Experience.

One advantage of the Bailey-Basili meta-model is that it identified many environmental attributes as possible contributors to effort. One disadvantage is that a local META model requires local data from past projects and results are difficult to calibrate.

2.3.1.7 Function Point Estimation Models

Function point estimation, developed by Albrecht (Albrecht, 1979) and later revised by the International Function Point User Group (IFPUG) (ISBSG, 2003) is a method for estimating effort through measuring the functionality of a system rather than its size. Longstreet (2005) states that function point estimation is a method to break systems into smaller components, so they can be better understood and analysed. It also provides a structured technique for problem-solving.

The total number of function points depends on the counts of distinct types in the five following classes:

- User-input types: data or control user-input types,
- User-output types: output data types,
- Inquiry types: interactive inputs requiring a response,
- Internal file types: files (logical groups of information) that are used and shared inside the system,

- External file types: files that are passed or shared between the system and other systems (Albrecht, 1979; Snell, 1997; Longstreet, 2005).

Each of these types is individually assigned one of three complexity levels where {1 = simple, 2 = medium, 3 = complex}, and at the same time, given a weighting value that varies from 3 (for simple input) to 15 (for complex internal files).

One advantage of this model is that data is available at an early stage of project development. It is also programming language independent. Finally, it is more accurate than the LOC estimate (Snell, 1997). However, some of the disadvantages of function points are that there is a need for subjective counting and evaluation, it is hard to automate, it ignores the quality of output, and it is oriented to traditional types of application (Futrell, Shafer and Shafer, 2002).

2.3.1.8 Checkpoint

Checkpoint is a knowledge-based software project estimating tool from Software Productivity Research (SPR) developed from Jones' studies (Jones, 1998). It has a database of thousands of software projects and it focuses on four areas that are used to ensure improvement in software quality and productivity. These four areas are technology, development process, environment and people management.

The checkpoint model uses Function Points (or Feature Points) as its primary input of size, and it focuses on three main parts for supporting the software development life-cycle. These three areas are Estimation, Measurement and Assessment (Boehm et al., 2000; Jones 1998). The advantage of this technique is simple techniques based. This approach is independent of technology and is a more accurate estimator in early phases. It also gives a complete function point counting and estimating capability and supports the calculation of Feature Points. The disadvantage of these models is that precision of the models is considered poor in predicting the effort.

2.3.1.9 PRICE-S Model

This model was originally developed by Price Systems at RCA to be used internally on their software projects (Freiman and Park, 1979). It was developed in 1977 by Freiman and Park, and used for estimating US Department of Defence (DoD), North America Space Agency (NASA) and other government software projects. The PRICE-S Model consists of three submodels that enable estimating costs and schedules for the development and support of computer systems. These three submodels are: the Acquisition Submodel, the Sizing Submodel and the Life-Cycle Cost Submodel. The above submodels allow for estimating costs and schedules according to project size, type and complexity (Park, 1988). Some of the advantages of the model are that the size of the software may be input directly, or automatically using function point sizing. It can also be customized based on the needs of the user and permits specific comparison. The model's main disadvantages are that it is difficult to estimate early in the life-cycle, the counts vary by language and many costs are not considered (e.g. requirements) (NASA JSC, 2002).

2.3.1.10 ESTIMACS

The ESTIMACS model (Rubin, 1983), offers a solution to problems relating to business specifications. It provides estimates relative to the six dimensions of effort hours, staff, cost, hardware requirements, risk and portfolio.

One advantage of this model is that the algorithmic system used at an early project stage is more accurate. One disadvantage is that the manner in which ESTIMACS translates input into effort is not entirely clear (Heemstra, 1992).

2.3.1.11 SOFTCOST

The SOFTCOST model for software estimating (Tausworthe, 1981) was originally developed for NASA in 1981. This model was improved using models and work carried out by Boehm, Putnam, Walston-Felix and others. The model uses LOC or function points as the primary input tool. A variety of the Softcost models were developed by Reifer Consultants, Inc. (Softcost-R, Softcost-Ada, Softcost-OO) from the work of Tausworthe (Reifer et al., 1999).

The Algorithmic models for cost estimation have their own challenges due to the fast changing nature of software development, which makes it very difficult to develop models that yield high accuracy in all domains. Models can be inaccurate if not properly calibrated and validated; moreover, it is possible that historical data used for calibration may not be relevant to new programs.

2.3.2 Expert Judgement Models (EJM)

Expert Judgement (non-algorithmic) models depend on the use of estimations based on prior knowledge and experiences. The expert techniques discussed below include the Delphi Approach and Work Breakdown Structure. These methods are based on the premise that where there is a lack of concrete and tangible data, experts may be consulted for their opinions (Boehm, 1981). These experts then supply estimates using their own experiences and knowledge of prior projects and relate them to the project at hand (Boehm, 1981). This provides an estimate of the cost, schedule, quality, etc. of the software being developed. The unmistakeable disadvantage of this method is that it is based on subjective opinion; and the opinion of one expert, may contradict that of another expert (Hughes, 1996). The Delphi approach and Work Breakdown structure are examples of expert judgement techniques. The following section describes each of them.

2.3.2.1 The Delphi Approach

The Delphi approach, originally from Greek oracle mythology, originated at the Rand Corporation in 1948, (Kaplan et al., 1949) as a way of making predictions about software estimation (Gordon and Helmer, 1964). It is used as a valuable way of getting group consensus on future software specifications. By getting group consensus on estimates, it eliminates the problem of individual, biased opinions (Snell, 1997). In order to improve and refine the estimate consensus obtained by the Delphi technique, Boehm and Farquhar created a new method which they called the wideband Delphi technique (Farquhar, 1970; Boehm, 1981).

An advantage of the Delphi approach is that estimation is subjected to a lot of discussion before coming to an agreement. Meetings allow people to express opinions and apply their own expertise. Another advantage is that if procedures are followed properly, bias can be eliminated and a group of experts come up with the best estimate for the project (Farquhar, 1970; Snell, 1997). Another advantage of the approach is that it can be more accurate if the system has been designed in detail. In addition, it is simple, inexpensive, utilises expertise of a number of people. A disadvantage of the approach however, is that it suffers from biases such as new manipulation of a group and silencing in order to see a specific outcome of a meeting. A further disadvantage of this approach is that it acquires lots of overhead costs in terms of time and person-hours for estimating some components of the entire project (Farquhar, 1970; Boehm, 1981; Snell, 1997).

2.3.2.2 Work Breakdown Structure (WBS)

As can be seen from the name, this technique breaks down the project in question into smaller components that can then be estimated separately. Once the smallest components are formed, estimates can be reached either by analogy or by experts. Once each individual component is estimated, the whole project estimate is derived by putting all estimates together (Boehm et al., 2000; Bunin, 2003). One advantage of the WBS-based technique is that it organises project elements into a hierarchical list of work activities, which simplifies the task of budget estimation. It is also good for planning and controlling costs, and measuring project progress. The main disadvantages to this method however, is that any estimate is only as good as the expert's opinion and it also encounters expertise-calibration problems. Finally, such a list is cumbersome to understand and may provide too much detail which is difficult to manage.

2.3.3 Case-Based Reasoning (CBR)

Case-based reasoning (CBR) is an approach which can be used to improve software effort estimation (Kadoda et al, 2000). CBR is a methodology for solving new problems by adopting the solutions of previous, similar problems. CBR is based on human reasoning; this gives a significant advantage in software development cost estimation since the results are intuitively acceptable by the people who make decisions. CBR retrieves existing cases, adapts them to a new situation, and finally generates solutions. This approach requires a large number of cases, which imposes high overheads when implementing a system (Aamodt and Plaza, 1994; Leake, 1996).

CBR depends on the availability of large databases of old software projects. Modifications can be made to adjust for minor changes in cases with similar attributes (Mair et al., 2000). CBR as a technique was explained by Aamodt and Plaza (1994) as a cyclical process made up of four stages as described by Kadoda et al. (2000):

- Retrieve previously similar cases related to current problem,
- Reuse the retrieved cases to find a solution,
- Revise the solution based on previous cases,
- Retain the new solution as a new case into the CBR database.

The process is also shown in Figure 2.1 below.

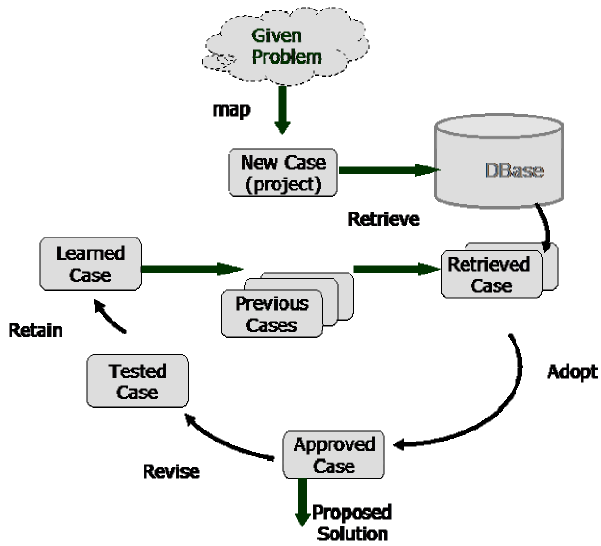


Figure 2.1: CBR cycle (adapted from Aamodt and Plaza, 1994)

The fact that CBR is derived from human reasoning has given this approach a significant advantage in software development cost estimation. CBR is simple and flexible compared to numerical models. Another advantage of CBR is that no expert is required because everything is based on the analogy taken from prior cases. Also, CBR can handle both quantitative and qualitative data. A disadvantage of CBR is that it needs a large volume of cases and it may be difficult to find cases with similar attributes. Another drawback is that predictions are limited to the cases which were found in the database; this can bias the resulting cost estimate (Snell, 1997).

Lederer and Prasad (1998); Shepperd and Schofield, (1997) discuss different methods of CBR and how they can improve cost estimation by transferring knowledge from past cases to new cases. However, CBR estimation accuracy still needs to be improved (Shepperd and Schofield, 1997). One approach might be to combine more than one estimation technique. For example, analogy might be used to help estimators take responsibility for the accuracy of their estimates by using performance reviews of systems. Local calibration is also necessary for consistent quantification and to minimise subjectivity (Lederer and Prasad, 1998; Delany, 1998).

2.3.3.1 Estimation by Analogy

Analogy involves comparing one or more completed projects in a similar domain as a means of producing new estimates. Estimation by analogy is a very simple, but powerful method. It has been used in many different applications and particularly in classification tasks (Shepperd, et al, 1996; Stamelos and Angelis, 2001). A certain number of nearest neighbours are defined according to a distance metric and their corresponding output values are then used to produce an approximation. The estimation of the outputs can be calculated by using the average of the outputs of the neighbours (analogies).

Hence, the analogy method can be described as a process with three steps:

- the new case for which the project effort is to be estimated is characterised by a set of common attributes,
- one or more similar cases (neighbours or analogies) from the dataset are found according to a predefined distance metric,
- the values of the neighbour cases are used to produce an estimate.

An important decision for analogy is the number of analogies (usually a small number) that need to be combined in order to evaluate the dependent variable. There are various approaches to selecting the distance metric and the number of analogies (Shepperd, et al, 1996; Stamelos and Angelis, 2001).

Estimation by analogy is a very simple, yet powerful method. It has a wide range of applications; particularly in classification tasks. Moreover, it is an attractive technique to apply to cost estimation, as it uses past experiences to solve new problems and corresponds to how experts operate (Delany, 1998). An advantage of this model is that it can be used with very few parameters in the early stages. A disadvantage is that, in order to compare projects, similar projects must exist. Also, there are difficulties with application domain dependence.

2.3.4 Hybrid Estimation Models

The use of a combination of techniques known as hybrid estimation models, may provide a good estimate. For example, combining algorithmic estimation with expert judgement and analogy methods may provide a superior estimation (Leung and Fan, 2003). Hybrid models tend to use both algorithmic and non algorithmic approaches.

2.3.4.1 Dynamics-Based Technique

The dynamics-based technique was developed as a model for software cost estimation which allows for changes over time. Forrester (1968) was the first to work on systems dynamics, formulating models that use continuous feedback loops. Madachy (1994) developed a dynamic inspection-based technique where the life cycle of the software process is supported by a continuous evaluation process. The system dynamics approach involves the following concepts (Richardson, 1991):

- defining problems dynamically, in terms of graphs over time,
- striving for an endogenous, behavioural view of the significant dynamics of a system,
- thinking of all real system concepts as continuous quantities interconnected in information feedback loops with circular causality,
- identifying independent levels in the system and their inflow and outflow rates,
- formulating a model capable of reproducing the dynamic problem of concern by itself,
- deriving understandings and applicable policy insights from the resulting model,
- implementing changes resulting from model-based understandings and insights.

An advantage of systems dynamics techniques is that they can be used to improve software cost estimation over time. One disadvantage however, is that the system can be complex and may grow into something that is difficult to handle, manage and calibrate (Abdel-Hamid and Madnick, 1991; Richardson, 1991).

2.3.5 Summary of Models

Table 2.1 presents a summary of the major features of various cost estimation models (circa 1960–1990). These models were chosen for comparison purposes because they each had different and distinguishable features and approaches. For example, some measure size of project through function points while others are measured by line of code. These models were also chosen, based on their leading techniques and relative strengths. No single technique is 'best' for all circumstances; the comparison and integration of models, results and approaches are the most likely way to produce better estimates.

Table 2.1: Various Cost Estimation Models - Activities/Factors/Approaches

Measuring Project size		SLIM (Putnam)[1]	Bailey-Basili[3]	Checkpoint[4]	ESTIMACS[5]	PRICE-S[6]	SEER-SEM[7]	Softcost-R[8]	COCOMO I & II[9]	Delphi[10]	System Dynamics[11]	Case-Based Reasoning[12]
Project's Technical Environment												
Effort	Skill (experience)	P	F	F	P	F	F	F	F	F	F	F
	Duration	F	F	F	F	F	F	F	F	F	P	F
Budget		P	N	N	F	P	N	N	P	N	N	N
Functionality		P	F	F	F	F	F	F	F	F	F	F
Lines of Code (LOC)		F	F	N	N	F	P	F	F	N	N	F
Function Points (System Requirements)		F	F	F	F	F	F	F	F	N	N	F
Feature Points (User Requirements)		F	F	F	F	P	P	F	N	N	N	P
Number of bubbles (details) on a data flow diagram (DFD / UML)		N	N	F	N	P	N	N	N	P	N	N
Number of entities on entity relationship diagram (ERD)		N	N	P	N	N	N	F	N	P	N	N
Count of process / control boxes on a structure chart		N	N	P	N	N	N	N	N	N	N	N
Number of objects classes, attributes & services / methods		P	N	F	N	F	N	F	F	N	N	N
Programming Language		F	N	F	N	F	F	F	F	F	P	P
Technology used (Hardware)		F	N	P	F	F	F	N	F	F	P	F
Software requirement (TOOLS)		F	N	P	F	F	F	N	F	F	P	F
ORGANISATION'S ENVIRONMENT												
Application Type		P	F	F	P	P	F	F	F	F	F	P
Organisation Type		P	F	F	P	P	F	P	F	N	N	P
Organisation's Line of Business		P	N	N	N	N	P	N	P	P	N	N
Organisation Culture & Project Leadership		N	N	N	N	N	N	N	N	N	N	N

F = fully supported (explicitly considered)
P = partially supported (partially considered)
N = not support (not able to determine from the available literature whether or not supported)

Footnote: for further references on these models see:

SLIM [1]:http://www.qsm.com/slim_estimate.html
Jensen[2]:http://sern.ucalgary.ca/courses/seng/621/W98/hongd/report2.html
Bailey-Basili[3]: http://sern.ucalgary.ca/courses/seng/621/W98/johnsonk/cost.htm
CheckPoints[4]: http://www.spr.com
ESTIMACS[5]: http://www.ca.com/products/estimacs.htm

PRICE-S[6]: http://www.pricesystems.com/
SEER-SEM [7] http://www.galorath.com/
Softcost-R[8]: http://www.lawsoncomputing.com/pceh-F.html
COCOMO[9] http://sunset.usc.edu/COCOMOII/cocomo.html
Delphi[10]: http://www.ecfc.u-net.com/cost/delph.htm; sern.ucalgary.ca/courses/seng/621/W98/johnsonk
System Dynamics Approach[11]: http://www.albany.edu/~gpr/
CBR[12]: http://www.ecfc.u-net.com/cost/index.htm
The International Society of Parametric Analysts (ISPA): http://www.ispa-cost.org.

2.4 Leadership and Organisational Culture

Leadership and culture are important issues in the success of any software project. Research has proven that leadership makes a huge difference in an organisation (Gardner, 1990; Swigger et al., 2003; Schneider and Barsoux, 2003). Culture and the approach of the leader affect what decisions are taken and in which direction the organisation is headed (Schein, 2004). Leadership and culture can also affect the way projects are run. For example, (Boehm, 1981) considers that the quality of a software team (i.e. the capability of programmers and analysts) is a significant factor in determining cost and quality in a software project (Krishnan, 1998). Gardner (1990), Sarros et al. (2006) and Fairholm (1991) noted the importance of leadership on the culture of an organisation. An organisation will manage more productively as a whole when its values are shared by its employees. Furthermore, organisational culture incorporates a set of assumptions, beliefs, and values, which guide organisational function (Hofstede and Hofstede, 2005; Schein, 2004). One would expect therefore that personality traits, power relationships, behavioural changes and organisational values will affect project costs.

2.4.1 Leadership

Leadership is defined by different people in different ways. Gardner (1990) sees leadership as a process of influencing others to achieve a task by providing purpose, direction and motivation. The main goal behind leadership is to follow a set of rules that steer the organisation in one direction or another (Fairholm, 1991).

There are a plethora of publications on leadership (Adair, 1998; Brown, 1998; Schein, 2004; Watson and Gallagher, 2005; Hunt and Larson, 1975; Hofstede, 1991; Schneider and Barsoux, 2003. However, comparatively little research has focused on leadership values (Fairholm, 1991). There have also been no studies to address the significance of leadership characteristics and organisational culture with regard to software cost estimation (Boehm, 1981; Kitchenham and Taylor, 1984; Shepperd et. al, 1996; Stamelos and Angelis, 2001).

Six key leadership parameters were derived from the literature review (Adair, 1998; Fairholm, 1991; Schneider and Barsoux, 2003), and from interviews with administrators, managers and project leaders (see Appendix C). The six parameters are:

- Interaction and relationships
- Decision-making
- Ability to motivate
- Understanding organisational culture
- Active thinking
- Communication skills

Each of these are discussed in the sections which follow.

2.4.1.1 Interaction and Relationships

According to Adair (1998), it is important for a leader to create learning experiences, pay personal attention to team needs and treat team members with respect. Members of the team have a need for interaction and involvement and understanding of those needs is important. Building good relationships with team members will strengthen loyalty and build responsibility to overcome various work situations. Balancing responsibility to the organisation with team relationships is complex; in some cultures managers tend to focus on the task and keep personal relationships aside (Schneider and Barsoux, 2003).

2.4.1.2 Decision-Making

According to Cosgrave (1996), decision-making is based on the knowledge, experience and constraints in place when a decision is made. These constraints can include pressure of time, availability of information and complexity. Time for consultation, reflection and discussion is also necessary (Schneider and Barsoux, 2003). Discussion with others can help in dealing with important decisions since problems may be resolved when discussed and evaluated, and positions shared (Hills, 2004).

2.4.1.3 Ability to Motivate

An organisation needs to motivate staff in order to achieve maximum performance. By enabling the participation of employees in a process and developing a dialogue between the team and the leaders, organisational values can influence the whole motivational process (Osteraker, 1999). This can result in reduced cost, savings in time, and more motivated staff (Osteraker, 1999; Fairholm, 1991).

2.4.1.4 Understanding Organisational Culture

It is important for team members to be involved in the decision-making process and to come up with suggestions to improve their work environment. Effective leaders often develop a strong team culture. The leader must actively work to enhance cultural understanding and mutual respect (Allen and Kraft, 1987). By doing this, the leader actively links and connects points of interests with the team members, thus fostering understanding and collaboration.

2.4.1.5 Active Thinking

It is important to build inclusive environments that enable leaders and facilitators to promote collective thinking. Effective problem-solving requires active thinking (Allio, 2006). Practising thinking aloud makes for strong relationships between workers, this in turn impacts on the quality of thinking. Thinking together as a team is very important to enable the different people involved in the organisation to achieve better results (Ringer, 2007).

2.4.1.6 Communication Skills

Good communication creates healthy environments where individuals can enjoy their work. Leaders must build relationships with both individuals and groups. Good communication involves leaders communicating with team members without showing favouritism towards specific groups (Harold, 2006). Creating dialogue, communicating objectives clearly and regularly, and facilitating good communication between team members are essential tools for leaders.

2.4.2 Organisational Culture

Culture plays an important role in people's lives in general, and in organisations in particular. Culture entails elements that come together to form the person or the society. It comprises factors such as knowledge, beliefs, values, traits, experiences, language and religion that make up a community, lifestyle and its way of thinking (Schein, 2004). Culture is also the concept used to express "the values, attitudes, and patterns of behaviour that are transmitted to all individuals in a particular social environment. Culture moulds the self, prescribes the relationships, and defines and reinforces our thoughts and feelings" (Poole, 2006).

Culture plays a significant role in helping individuals enhance their learning (or hinder it). Numerous authors (Gardner, 1990; Swigger et al., 2003; Schneider and Barsoux, 2003; and Schein, 2004) have written about culture and its effect on the organisation's approval for learning. Team performance, behaviour and attitudes within the organisation and outside are affected by culture (Schein, 2004). Hence it is necessary for project managers and leaders to promote a positive, inclusive culture.

The seven cultural parameters outlined below were derived from the literature review and research conducted at UAE University in 2007 (Abu-Rmaileh and

Hamdan, 2007). According to the literature review, they were the most important parameters issues in the fields of education, sociology, leadership and administration. The seven parameters are:

- Timeliness
- Collaboration
- Job stability
- Intercultural intelligence
- Team experience
- Communication
- Reward mechanism

Each of these are discussed in the sections which follow.

2.4.2.1 Timeliness

Time awareness at work is a significant factor. There are issues around how different cultures value time and successfully use time at work. For an individual how they manage their time and balance the different tasks that they are doing is important. A successful team realises the value of time; team members that struggle are usually the ones that let time slip by and do not value time as they value money (Crawford, 2007). A person may have the right to waste his own time, but not that of others.

2.4.2.2 Collaboration

Collaboration is a dynamic process that has to be built using connections between individuals in a group. Building collaborative relationships is a long-term process and does not end after the successful completion of a project. Collaboration works best when everyone is involved; taking risks in both personal and organisational dimensions (Solomon et al., 2001). Teamwork performance is often evaluated with respect to team members' attitudes about how well the team collaborated, regardless of the success or otherwise of the project (Elfenbein and O'Reilly, 2002).

2.4.2.3 Job stability

It is difficult for individuals to work in an environment where there is no job security. Sadri (1996) shows that one potential outcome of the lack of job stability may be reduced productivity. Effort, ability and morale are all factors that determine a worker's productivity. Organisations with low job security may experience increasing tardiness, absenteeism and reduced turnover (Sadri, 1996). When people lack job security, they tend to have low morale and give up or become lax in their work.

2.4.2.4 Intercultural Intelligence

In general, emotional intelligence is a concept which refers to the ability to be aware of one's feelings and the feelings of others, to differentiate among them and use the information to guide one's own thinking behaviours (Osborn, 2006; Rahim and Minors, 2003). This will have an impact on staff performance and satisfaction. Intercultural intelligence is the ability not only to understand cultural differences, but also to adapt one's behaviours appropriately in order to fit in the other culture. Thus, it is important for business success. High social-emotional intelligence shows that a person performs well both as an individual and with others (Gabel et al., 2005). Good teams are those that are flexible, adaptable to new situations and are always working in the best interests of the team.

2.4.2.5 Team Experience

Individual experiences affect teamwork, because each person comes with a different set of skills gained from a different organisational culture. Team backgrounds often provide better work experiences and outcomes (Hartenian, 2003).

2.4.2.6 Communication

In general, communication includes all verbal and nonverbal behaviour between people, including language, thoughts and feelings, problem solving, and learning. It also reveals how issues of identity, interdependence, power, social distance, conflict and negotiations are managed (Schneider and Barsoux, 2003). Miscommunication is a subtle problem that results from cultural differences in the meaning of verbal and nonverbal behaviour (Poole, 2006). This also includes effective listening so that information is not misunderstood.

Effective teams are good at communicating with each other (Ryder-Smith, 1999). Furthermore, when communication is not working well, this will have a negative impact on the team. When a team does not communicate effectively it will lead to a loss of time and productivity.

2.4.2.7 Reward Mechanism

Rewarding individuals is one of the most important factors that affects team motivation. A good reward scheme can increase production and lower costs. Rewards will inspire team members to be involved in positive change towards continuous improvement. It also aims to reward those individuals that provide examples of desirable behaviour and achieve the organisation's objectives (Stewart, 1989).

2.5 Summary

Estimation methods fall into three main categories: algorithmic cost estimation, expert judgement, and case-based reasoning. All three approaches have advantages and disadvantages. This chapter has reviewed current methods. The major limitations of the current methods are:

- inaccuracy or lack of success in making estimations,
- extreme difficulty in forecasting,
- limited information available at early stages of a project,
- little available domain knowledge,
- no single method is the best for all projects.

This chapter has reviewed the literature and identified the important factors within leadership and culture. The literature review has shown that leadership and culture go hand in hand. They are also important factors in the way an organisation functions. An investigation by survey will determine whether leadership and culture are also important factors for successful cost estimation of software development. The factors which will be explored are:

Leadership:

- Interaction and relationships
- Decision-making
- Ability to motivate
- Understanding organisational culture
- Active thinking
- Communication skills

Culture:

- Timeliness
- Collaboration
- Job stability
- Intercultural intelligence
- Team experience
- Communication
- Reward mechanism

These areas are the basis for further survey work in the next chapter.

CHAPTER 3

The Surveys

3.1 Introduction

This chapter describes the process of questionnaire development, survey implementation, data collection and analysis. A staged approach, in three phases, was taken. This consisted of a pilot study and two main surveys. The aim of the project is to investigate the cost estimation models in the UAE and the Gulf States. Consequently, government departments and oil companies in the UAE were selected to participate in the survey. The selected government departments represent 70% of the total budget in the UAE. Oil companies were selected because they represent the most mature private and semi-private organisations. A list of all the government departments (federal and local) and the oil companies was compiled, then those departments that have major IT initiatives and mature budgets were short-listed. Moreover, only government departments which were willing to play an active role in the survey and support this research were retained. Aside from the government departments the sample included a wide range of different service providers (Municipalities, Civil Defence, Water and Electricity, and Education). The selected government departments represent 52% of the total number of government departments in the UAE. All short-listed government departments were surveyed. In each of the surveyed organisations, the project leader(s) and team members were selected for interview and were also asked to complete questionnaires (see sections

3.3.4 and 3.3.5). The total number of organisations involved 24 project leaders and team members with 41 projects. These comprised the unit of analysis.

Initially, a pilot study (see section 3.6) was undertaken to test the questionnaire and to perform an initial investigation into the methods that government departments use to estimate the cost of software development. A first version of the questionnaire was prepared in an effort to determine the strategies used by government departments in Abu Dhabi, the capital city of UAE, for software development estimation.

The questionnaire was then revised (see Section 3.7) to better serve the objectives of this research. At this point the survey was expanded to cover the Dubai emirate, with the aim of solidifying the key findings and including more advanced IT projects. The questionnaire was also modified at this stage to capture data on leadership and culture.

The third and final survey (see Section 3.8) included further attributes based on the hypothesis and also included a structured interview. The main objective of this survey was to further investigate the impact of leadership and organisational culture and further investigate the role they play in software development effort estimation.

3.2 Overall Aims of the Survey

The aims of the survey were:

- To identify and analyse approaches that government departments follow in managing software development and maintenance within organisations in the Gulf States.
- To identify the needs of the Gulf States software development community with respect to software project estimation.
- To determine the needs for a model for software cost estimation in governmental departments.
- To determine the impact of organisational culture and leadership on software effort estimation.

As indicated in the literature (Madar Research Group, 2003; Hamdan et al, 2005), the approaches used for software cost estimation may differ and can be affected by many factors, such as the cultural environment, political system, leadership style and economic strength. This research assumes that the Gulf States are similar in their approaches to software development.

3.3 Questionnaire Structure

As stated earlier, the questionnaire was refined in three versions:

- The original version of the questionnaire was based on the work of Lederer and Prasad (1992) and Futrell, Shafer and Shafer (2002), who investigated the process used to estimate the effort of developing and maintaining Information Systems in government and semi-government departments in Abu Dhabi. The objectives of this study were to determine the current state-of-the-art software effort estimation techniques to help identify methods and research directions for improving effort estimation and control (see questionnaire in Appendix A). The questionnaire contained a set of topics and questions that were intended to gather information about specific projects which have been undertaken by the organisations.

- The second version of the questionnaire was expanded to include factors not covered by the first version. For example, it was found that person-days and budgets are the most popular estimation measurement, and these were therefore included to define the size of project (see questionnaire in Appendix B).
- The third version of the questionnaire was expanded to include cultural and leadership issues. It was also much more detailed than the other two. For instance, questions were added enquiring about operating systems, databases, etc. (see questionnaire in Appendix C).

Each of these versions will now be explained in detail below.

For the Pilot Study, 11 government departments in Abu Dhabi were selected. They were asked to provide information regarding the software projects they had been developing. This study focused on collecting related information as to the type of software systems developed, as well as data on variances between actual cost and estimated cost. The different departments were asked to provide information regarding the software projects they had developed and whether most of the effort was expended on minor or major upgrades of systems or developing new systems. The information requested included cost, quality and planning methods. The information was collected and validated over a three-month period.

The Pilot Study showed that none of the projects were delivered successfully. The projects were either over-budget, missing some functionality or had missed deadlines. The reasons were mainly due to failures in effort estimation. Even though some of the team members had previous experience of effort estimation, they were not formally trained in cost estimation. It was also found that most project managers preferred to over-estimate cost to be on the safe side, and so higher management would not blame them for failure.

For Survey 1 the questionnaire was modified, more attributes were added and 20 government departments were studied. Survey 1 was applied to government and semi-government departments in Dubai and Abu Dhabi. Data were collected and validated over a six month period. The process of collecting information in Dubai was relatively fast compared to Abu Dhabi, because as stated earlier, the Dubai government has more well-defined IT strategies.

Survey 1 produced a recommendation that a national database be built to store all the historical lessons learned from previous government projects. Such a database would be a great asset, since most software development projects in government departments are similar. The lack of historical data and models has impacted negatively upon the effort estimation process.

Major discrepancies existed between estimated and actual project cost. There was also a lack of training. In contrast to Abu Dhabi, it was noticed that Dubai has more advanced Internet applications and ERP systems. However, similar to Abu Dhabi, the process of cost estimation was still immature and lacked the use of any formal model.

It was also noticed that person-days and budgets were the primary parameters used by government departments to define the size of project. Software functionality was also used to determine the size of projects. Therefore the third questionnaire was modified to capture information on these two issues in Survey 2. A summary of the questions asked is given below.

3.3.1 Organisation's Line of Business

This section identified the organisations' line of business. Types of business were Education, Health, Military, Civil Service, Oil-Gas, Tourism Services and Telecommunications.

3.3.2 Application Type

These were categorised into two areas: core systems and supporting systems. Each is explained in more detail below.

Core systems exist to achieve the core mission of organisations and to satisfy their core purpose. Examples of core systems are:

- Fire alarm systems used by civil defence departments,
- Flight information systems used by civil aviation departments,
- Traffic light management systems in police and traffic departments.

Supporting systems support the internal services of an organisation. Examples of supporting systems are:

- Human resources and payrolls,
- Financial systems,
- Document management systems.

3.3.3 Organisational Structure

Organisations were asked to classify themselves according to three possible structures (Futrell, Shafer, and Shafer, 2002): project oriented, functional and matrix. In a project-oriented (projectized) organisation, the project manager has the highest authority. In this type of organisation a project establishes a unity of command. All skills needed are assigned to the project. In a functional organisation structure, the project manager has less authority and staff are divided into their functional specialities (e.g. software engineers would report to the engineering department manager). Project managers basically focus on key functions and concentrate on core competencies. In a matrix structure, the project manager and the functional manager share similar authority. There is a balance of power established between them. The characteristic of a matrix organisation is that it enables project objectives to be clearly communicated and a functional disciplinary team is retained. It may create duplication of effort across projects. Also, functional and project management may have different priorities.

3.3.4 Project Leadership

While studying the parameters that impact the software effort and cost estimation in government departments, it was concluded by the author that leadership characteristics should be considered. For example in Abu Dhabi and Dubai Project managers tend to have more authority (project-oriented). Also, project managers in Dubai tend to have their own distinct style. The following sections describe the attributes that were considered in this questionnaire.

a) Interaction and relationships

It is important for a leader to create learning experiences, pay personal attention to team needs and treat team members with respect. Team members have a need for interaction and involvement and understanding those needs is important. It is important for the leader to pull the team together by asking them for their feedback and ideas about how things can be made more effective. Good communication will strengthen relationships and improve performance. The following questions were asked to examine this area:

- Does the leader pay attention to what the team members have to say?
- Does the leader listen, consider proposals and allow others the freedom to express feelings?
- Does the leader support the team when necessary?
- Does the leader give individual and team feedback on performance in an honest manner?
- Does the leader focus on human values?
- Does the leader care about the team?

b) Decision-Making

The leader must hold effective meetings with a focus on decision-making, and make appropriate decisions by consulting with the team. Decisions should be made by general consensus which empowers others to fulfil requirements. The following questions were asked to examine decision-making:

- How often do leaders make decisions?
- What types of decisions does the leader take (hiring / firing, etc.)?
- How does the leader communicate critical decisions (face-to-face, e-mail, memos, etc.)?

- Does the leader consult with team members before a decision is made?
- How decisive is the leader in taking decisions?
- Are the decisions made by the leader in a timely manner?
- Does the leader delegate decisions and jobs to others?
- How creative are the decisions that the leader makes?

c) Ability to Motivate

It is important that the leader has the ability to motivate team members to fulfil goals, meet targets and share goals and visions with team members. The leader also needs to provide appropriate levels of direction and support to enhance performance. Most organisations make an attempt to motivate their employees by making use of rewards. Leadership effectiveness is measured in terms of how well the team accomplishes its goal, and how successful the leader is in motivating behaviour. The following questions examined this particular attribute:

- What motivational approaches are used?
- What does a leader actually do to inspire the team?
- What basis of influence can a leader use?
- How does the leader keep team members satisfied?
- What behaviour is rewarded?
- Are values backed up by time and money?

d) Understanding Organisational Culture

The leader must understand and actively work to enhance cultural understanding and respect among others. Also, the leader must effectively articulate the mission (or vision) with clarity and respect and demonstrate the ability to understand and manage intercultural teams so that everyone conforms to the majority. The following questions were asked to examine this particular attribute:

- Does the leader understand and manage the intercultural team?
- Does the leader enhance cultural understanding and respect among the team?
- Does the leader believe everyone should conform to the majority?

e) Active thinking

The leader must develop a clear vision of the project and set achievable targets. The leader must also see that the visions, values and goals of the team are in line with those of upper management, in order to enhance creativities and team contributions. The following questions were asked to examine this particular attribute:

- Does the leader enhance team contributions and set feasible targets?
- Does the leader see that visions, values and goals of the team are in line with management visions?
- Does the leader engage with ideas and concepts to know the big picture?
- Does the leader give feedback constantly, irregularly, at job completion, or never?

f) Communication Skills

A leader should carry out enhanced communication as well as creating dialogue and channels in and through which that communication takes place. Building effective communication skills is an important key that can open closed doors. The leader must communicate objectives clearly and regularly with team members and assist in good communication among team members. This includes communicating effectively for understanding and empowerment during the change process. The leader must also create a positive atmosphere in the work environment and address work related issues clearly, justly among the team and give effective feedback on team related issues. This process produced the following questions to examine the communication skills attribute:

- Has the leader communicated the vision, values, goals and rules of the institution with the team members?
- How often does the leader communicate with his team members?
- Is the leader clear in the way he communicates with the team members?
- What way does the leader use to communicate with the team members (face-to-face, e-mail, memos, etc.)?
- How is the information conveyed (directly or indirectly, through assistants, or by themselves)?
- How clear is the communication that the leader conveys?
- Does communication take place in an open forum or is it one-sided?
- Does communication take place in a timely manner?

The impact and the influence of these attributes were studied in detail, as leadership and the project organisational structure were considered to be an important part of effort estimation. The leadership assessment part of the questionnaire was a refinement of a model of leadership and character developed by Fairholm, (1991); Schein, (2004).

3.3.5 Organisational Culture

Previous studies have ignored the impact of organisational culture in determining software development effort. This research has considered the importance of the organisational culture. Towards this end, several parameters were included. Each of these parameters are explained below.

a) Timeliness

Team members' respect for time and commitment to time were investigated. The following questions were asked to examine this particular attribute:

- Does the team manage time and resources in an effective manner?
- Do team members respect time and value it?
- Do individuals understand and adapt to the team's general view of time?
- Does the team maintain a balance between work and social life?

Time awareness in work and outside work is a significant indicator as to how individuals view time and how they utilise it effectively. It is also an issue of how teams in different cultures value time and how successfully they use that time.

b) Collaboration

Team members were also asked about their relationships with each other. Cooperating, trusting or information-seeking/supporting and constructive team relations were studied and measured in this survey. The significance of collaboration is for teams to realise that they can match their interests and are able to benefit from working with each other. The following questions were asked to examine this particular attribute:

- Do individuals work with each other to enhance team commitment?
- Does each individual understand the work of others?

- Do individuals have personal contact and provide feedback to others?
- Do team members have the right attitude and perspective?
- Do team members have positive peer relationships?

c) Job stability

This reflects loyalty to the project and the organisation. Team loyalty, belonging and trust among the different team members, were considered in this survey. It is important that the team and the leader develop a positive relationship with mutual trust. The following questions were asked to examine this particular attribute:

- Does a team build relationships and develop trust among others?
- Does an individual enhance team loyalty and belonging to the group?
- Does a team take risks and feel secure enough to achieve goals?

d) Intercultural intelligence

Impersonal relations were determined for each project. The ability to understand and respect the other's culture or point of view was investigated. The following questions were asked to examine this particular attribute:

- Do the team members who are responsible for success encourage others?
- Does the team promote creativity of its members?
- Does the team understand other cultural views?
- Does the team recognise the strengths and abilities of others?

e) Team experience

The skills of the team members were investigated and studied. These were considered important when determining the similarities among organisation culture. The following questions were asked to examine this particular attribute:

- Does the team use tools and equipment effectively?
- Does the team use skills and knowledge acquired for the project?

- Are team members familiar with other members' backgrounds and experiences?
- Does the team find solutions to problems in order to maintain progress?

f) Communication

It has been identified that communication plays a major role in project success or failure. The following questions were asked to examine this particular attribute:

- Does the team assist and help both tasks and personal issues?
- Does the team give and receive feedback on their teamwork?
- Does the team consciously reflect on their effectiveness and strive for improvement?

g) Reward mechanism

Three methods of reward were identified in the questionnaire. The following questions were asked to examine this particular attribute:

- Does the team encourage and support others in on-going professional growth opportunities?
- Does the team recognise and reward achievements?
- Does the team encourage the achievement of desired results?

3.3.6 Project Technical Environment

This section of the questionnaire covered parameters that impact upon software effort and cost estimation, namely:

- Number of core users (backend users): users who use the system for their day to day job function
- Number of clients: i.e. the set of users in each organisation
- Number of transactions: i.e. the set of actions and instructions
- Numbers of entities: objects, functions, and transactions

- Technology used (hardware and software infrastructures)
- Skill sets were considered in two dimensions: cost and experience

3.3.7 Year of Project Completion

The questionnaire asked the year of project completion. Comparisons were only to be made between those projects that were implemented during the previous five years.

3.3.8 Effort and Project Duration

In an effort to determine the validity of effort prediction the questionnaire measured the actual effort and the project duration.

3.4 Data Collection

The process of collecting the information was conducted in three stages each using a different survey. The Pilot Study was conducted in Abu Dhabi with 11 government departments. In Survey 1, 20 government departments were studied. Survey 2 was a comprehensive questionnaire that included all the important variables required to measure the effort estimation. In an effort to improve the accuracy of the collected data, visits to each department were also conducted.

Departments were selected on the basis of being the largest and oldest. More recent government departments may not have a mature IT infrastructure. The UAE was ranked #1 among the Arab countries in 2002 to 2004 for having the most advanced e-government (www.unstats.un.org; Global 2005). The selected departments represent 70% of the government departments in Abu Dhabi and Dubai and share 73% of the IT spending in UAE. Total spending of these departments for software development and information systems circa 2003 was approximately US $1,265 million.

The departments were surveyed over a period of nine months, over which time questionnaires were sent to each department. Twenty-four departments were selected for the study. In order to facilitate the process of collecting information, the questionnaire was designed in a multiple-choice format and completed during one-to-one meetings.

The survey was validated by physically visiting sites and examining information from previous projects. The websites of the departments were also visited to further validate the data collected. The questionnaires were initially sent via email to the participants. Interviews were then conducted with each participant to further explain the purpose of the survey and to ensure accuracy of the data. Each interview was less than one hour duration. It was made clear that the names of all participating organisations would remain confidential.

3.5 Pilot Study

The first study was a pilot survey to test the questionnaire and to perform initial investigations into the methods that government departments used to estimate the cost of software development projects. The pilot study also identified attributes to include in the later surveys, and help establish procedures for future survey work.

3.5.1 Application Areas

It was found that 55% of the departments developed their applications in-house while 36% outsourced their application development. Only 9% of the companies used both in-house and outsourcing. Table 3.1 illustrates this information. This may be because the concept of outsourcing is still new in the region, thus there are limited outsourcing options.

Developed methods	
In-house	54.5%
Outsourcing	36.4%
Both methods	9.1%

Table 3.1: The ratio of applications developed in-house, outsourced and both methods

ERP systems occupied the majority of IT applications (64% of the developed applications); this is due to the fact that most government departments started building ERP solutions in the mid 80s (see Table 3.2).

Developed application	
Enterprise Resource Planning (ERP)	64%
Decision Support Systems	10%
Process Control Systems	3%
Business Related Systems	15%
Other	8%

Table 3.2: The ratio between the types of the developed applications in the Abu Dhabi departments

Furthermore, it was found that 27% of the in-house applications were focused on developing fully functional application software excluding hardware and software licences. However, 73% of software development in-house included hardware and software on their budgets. This is due to governmental purchasing regulations. It is easier for a government department to include both hardware and software in the contract (see Table 3.3).

IT projects preference	
Project excluding Hardware & Software	27%
Project including Hardware and Software	73%

Table 3.3: IT projects preferred within the Abu Dhabi organisations

It was found that 58% of applications are new, 12% are major upgrades, 12% are minor upgrades, and 18% are updates of existing applications (see Table 3.4). As mentioned previously, many government departments built their ERP applications in the mid 80s and now wanted to renew their systems, rather than upgrade them.

The anticipated delivery of projects	
New system	58%
Major upgrade: a currently existing system was upgraded, requiring extensive amounts of new software to be developed	12%
Minor upgrade: a currently existing system was upgraded requiring (some) software	12%
Updated system by modifying existing software	18%

Table 3.4: Expected future projects

The majority of the government departments developed their own applications in-house and, only a small number of those departments paid attention to software development effort estimation. Departments are required to allocate their IT budget at the beginning of each fiscal year without focusing on software development effort estimation, due to the budgeting regulations. As shown in Table 3.5, most departments calculate the cost of the total project rather than focus upon software development effort.

The anticipated delivery of projects	
Estimated the software development effort	27%
Looked at the project as a whole	73%

Table 3.5: The departments anticipated in software development effort estimation

Most of the departments stated that there were discrepancies between estimated effort and the actual effort. As shown in Table 3.6, in 37% of cases the estimated effort was more than the actual effort, while, in 18% of cases the estimated effort was less than the actual effort. Only 27% felt that their estimations were acceptable. Most project managers preferred to estimate higher budget in order to be on the safe side.

The anticipated delivery of projects	
More	37%
Less	18%
Acceptable	27%
Not applicable (N/A)	18%

Table 3.6: The discrepancy between the estimated effort and the actual ones

The Pilot Study also investigated the reasons behind discrepancies between actual effort and estimated effort (see Table 3.7). Unanticipated tasks are the main reason for discrepancies, while the skills of team members, ignoring performance reviews and missing data, are also factors. These results demonstrate an urgent need for improved estimation methods.

Effort estimation discrepancy	
Inability to anticipate skill of project team members	8%
Overlooked tasks	20%
Lack of an adequate methodology or guidelines for estimating	12%
Lack of historical data regarding past estimates and actual performance	4%
Lack of project control comparing estimates and actual performance	8%
Frequent requests for changes by users	12%
Performance reviews which did not consider whether estimates were met	8%
Poor requirements (lack of cooperation and lack of experience)	4%
Missing Data	8%
Other	16%

Table 3.7: Reasons for the effort estimation discrepancies

3.5.2 Estimation Methods

Another reason for discrepancies between actual effort and estimated effort is that software development estimation is conducted in the early stages of a project. Project managers are obliged by regulations to set the software development budget prior to, or at the early stages of, the project launch.

The Pilot Study investigated the phase in which the estimation occurred. 34% of the departments stated that effort estimation occurred prior to the project award.

Table 3.8 below indicates whether estimation took place in the early stages of a project.

Effort estimation occurs	
Beginning of each phase	25%
Initial project plan	17%
Prior to project award (before approval)	34%
Prior to implementation stage	8%
After system requirement analysis phase	8%
End of stage 1 out of 4 stages	8%

Table 3.8: When effort estimation occurred during a project

The Pilot Study also asked the parameters that were considered in estimating software development. 36% included buildup of manpower (recruitment and retention of staff with essential skills), 14% resources needed, 14% IT Infrastructure (person-days, cost, equipment and training), 29% total effort, and 7% duration (seeTable 3.9).

Parameters considered for estimating software cost	
Buildup of manpower (specialists in profession)	36%
Resources needed	14%
IT Infrastructure (person-day, cost, equipment, and training)	14%
Total effort	29%
Duration	7%

Table 3.9: The factors that are considered in effort estimation

In order to gain revised and more precise estimation and to avoid any risks, project managers will often perform some informal cost estimations (Futrell, Shafer and Shafer, 2002). However, they will tend to reduce small elements of the project scope, rather than request a revision of the budget. Table 3.10 shows when re-estimation occurred during the project life-cycle.

Cost re-estimation	
Informal re-estimates performed during development.	8%
Formal re-estimates performed at pre-defined milestones/ targets.	38%
An amendment changed the system being built and a re-estimate required.	31%
Re-estimation was not performed	23%

Table 3.10: Re-estimation

Table 3.11 shows the parameters that were re-estimated.

Items included in the re-estimate	
Software	18%
Hardware	19%
Number of delivered units	9%
Testing	12%
Integration	14%
Documentation	12%
Training	14%
Other	2%

Table 3.11: Re-estimation parameters

3.5.3 Estimation Team

Project managers and many of their IT staff were involved in the estimation process (see Table 3.12). Despite this diverse allocation of personnel, only 5% of them had any previous experience in cost estimation.

IT Position	
C.E.O / President	--
Database Manager	3%
IT Consultant	13%
IT Engineer	3%
IT Manager	29%
System Analyst	6%
General Manager	3%
Development Manager	9%
Programmer	6%
IT Director	19%
Strategic Planning/Project Admin.	3%
Technical Advisor	6%

Table 3.12: IT players in cost estimation

Also, it was observed that only 40% of project managers had been formally trained (e.g. adequate training in risk identification and management, also adequate training on tools to manage cost estimating, control and reporting) (see Table 3.13). However, none of them had fully used their expertise in effort estimations or utilised cost estimation model.

Project Manager training	
Project manager took training	40%
Project manager did not join training	60%

Table 3.13: Project managers who attended training and those who did not

3.5.4 Lessons learned from Pilot Study

The pilot survey showed that none of the projects studied were delivered entirely successfully. It found that software development projects were either over-budget, had missed deadlines or fell short of some of their required functionality. The reasons behind this are mainly due to a failure in effort estimation. These results may be summarised as follows:

- Software cost estimation occurred at the early stages of the project and was consequently not based on a well-defined cost estimation model.
- People in charge of software development lack the experience and the skills in software cost estimation.
- Rules and regulations regarding information systems prevent or hinder budgetary control of the IT project.

It was noticed that the ERP systems occupied the majority of IT applications and most wanted to renew their ERP systems rather than upgrade them. It was also found that most project managers prefer to estimate a larger budget for a project in order to be on the safe side. Most importantly, there was a complete lack of a cost estimation model.

Implementing a database that stores historical data and lessons learned from completed projects would help greatly in achieving more accurate estimation; such a database would save the Gulf States a considerable amount of money.

3.6 Survey 1

Survey 1 was based on the pilot study and examined extra parameters used by government departments to estimate software cost. For example, the methods used to estimate project size; budget, person-hours or functionality. Survey 1 also extended the study to cover government departments in Dubai.

The pilot survey indicated that no formal model was used in Abu Dhabi for software cost estimation. This resulted in failures for most of the application development projects. It was concluded that the questionnaire needed to be revised to better serve the objectives of this research (see Appendix B). The results of Survey 1 are compared with those of the Pilot Study in the following sections.

Outsourcing software development is the dominant option that government departments in Dubai follow when building information systems. As illustrated in Table 3.14, 87% of government departments in Dubai have outsourced software development, whereas 13% develop their own applications. These findings suggest that government departments are focusing on their core business, and also point to the availability of multinational and regional software development companies located in Dubai.

Project developed	Abu Dhabi	Dubai
Within your organisation (in-house)	54.5%	13%
Contracted out (outsourced)	36.4%	87%
Both methods	9.1%	--

Table 3.14: The ratio of the developed applications in-house and outsourced

The main reason behind this discrepancy between Abu Dhabi and Dubai is probably due to the very limited outsourcing options and because the concept of outsourcing is still new in Abu Dhabi. In contrast, the option of outsourcing is widely available in Dubai (for example "Dubai Internet City" and Etisalat Telecommunications Co.) and the concept of outsourcing the application development and hosting is highly supported by top leadership.

The survey has shown (see Table 3.15) that government departments have paid similar attention to the development of enterprise resource planning systems, e-services, document management systems; intranet applications and help desk systems. This is according to the directions of the highest authority in Dubai to e-enable all services and to allow the public and businesses to interact with government services directly through online channels. (The Dubai leadership has set a target to e-enable 90% of government services by the year 2007).

Developed application	Abu Dhabi	Dubai
Enterprise Recourse Planning System (ERP)	64%	20%
Document Management Systems	15%	18%
Help Desk System	8%	18%
Decision Support System	10%	8%
Intranet Applications	3%	18%
eServices (automated services)	--	18%

Table 3.15: The ratio between the types of developed applications

It was found that 64% of the developed applications were ERP systems. This may be due to the fact that government departments in Abu Dhabi are focusing on automating their internal processes. The concept of providing electronic services to the public is still in the early stages in Abu Dhabi; this may be why major steps towards e-Government were not noticed. However, in the near future most government departments will launch projects to e-enable their services. ERP occupied only 20% of the overall developed applications. This may be due to the fact that most government departments started building their

ERP solutions in the early 1980s. Internet applications and e-services occupy the second largest developed IT applications; this is due to the Dubai leadership to e-enable all government services and because it has more development, free training zones, many international companies and is also an Internet city.

3.6.1 General Information

It was noted that more than 50% of the applications developed in Abu Dhabi were built from scratch or involved customisation, in contrast with Dubai where it is less than 40% (see Table 3.16). The main reason was due to the fact that Dubai government departments have actually started the utilisation of IT applications before Abu Dhabi. The other reason for this difference is because Abu Dubai government departments tend to allocate higher budgets for software development; whereas in Dubai, IT budgets for redeveloping applications are tighter.

Developed Project	Abu Dhabi	Dubai
Fully functional software developed from scratch	58%	39%
Readymade software with major customisation	18%	38%
A major upgrade – an existing system with extensive customisation	12%	15%
A minor upgrade – an existing system with few amount of customisation	12%	8%

Table 3.16: Developed projects

As shown in Table 3.17, both Abu Dhabi and Dubai are almost the same when it comes to defining the scope and the size of the project. This is due to the unified federal rules and regulations that manage the process of purchasing software. It is easier for government departments to include hardware and software in a contract to avoid complexity and to ensure that the budget is reserved for the newly-developed system.

Software development and implementation dominates governmental department spending in Dubai, consultancy services occupy 18%, while hardware purchasing and other information technology activities take up the other 18% of effort.

Contracted scope	**Abu Dhabi**	**Dubai**
Consultancy Services	27%	18%
Software development and Implementation	73%	64%
Hardware purchase and Implementation		9%
Other	--	9%

Table 3.17: Contracted effort on IT projects

As shown in Table 3.18, government departments in Abu Dhabi and Dubai prefer to allocate more budgets to bids to be on the safe side. This is because of the lack of appropriate and reliable models for software cost estimation.

Estimated efforts to bid contracts		**Abu Dhabi**	**Dubai**
> 50%	More	37%	29%
> 25%			14%
Same or close		27%	29%
< 50%	Less	18%	14%
< 10%			14%
N/A		18%	--

Table 3.18: The discrepancy between the estimated effort to the bid
Same or close: (Variance between actual effort and estimated is negligible)

Most of the departments interviewed stated that there was a discrepancy between the estimated effort and the bid contract. As indicated in Dubai, 29% of estimations differed by more than 50% compared to the original bid, and 14% of estimations differed between 25% and 50% compared to the bid; only 29% of estimations were close and acceptable. This finding is similar to those of the Pilot Study, which indicated that estimations are still weak in Dubai. This shows the need for a model for software cost estimation.

As shown in Table 3.19, the lack of a methodology for effort estimation, as well as the lack of historical data from past projects in Dubai represents 36% of the reasons for discrepancies between estimated effort and the bid proposal. This emphasises the need for a new cost estimation model.

Effort estimation discrepancy	**Abu Dhabi**	**Dubai**
Inability to anticipate skill of project team members	8%	4%
Overlooked tasks	20%	17%
Lack of an adequate methodology or guidelines for estimating	12%	18%
Lack of historical data regarding past estimates and actual performance	4%	18%
Lack of project control comparing estimates and actual performance	8%	13%
Frequent requests for changes by users	12%	20%
Performance reviews which did not consider whether estimates were met	8%	7%
Poor requirements (lack of cooperation and lack of experience)	4%	7%
Missing Data	8%	--
Other (Initial requirements were vague)	16%	6%

Table 3.19: Reasons for discrepancies between estimated effort and bid proposal

3.6.2 Estimation methodology used

Due to department polices and regulations, 75% of departments conducted software effort estimation prior to the project being awarded (see Table 3.20).

Stage of project effort estimation occurs	**Abu Dhabi**	**Dubai**
Prior to project award	76%	75%
After the Bids review and approvals	24%	25%

Table 3.20: The stage at which the effort estimation occurred during the project

Table 3.21 shows the re-estimate of effort performed during development; 36% of departments conducted effort re-estimation.

Were any re-estimates of effort performed during development	**Abu Dhabi**	**Dubai**
Yes	82%	36%
No	18%	64%

Table 3.21: Re-estimates of effort occurs

Table 3.22 highlights the reasons behind the cost re-estimation. If departments used cost estimation models, the need to perform cost re-estimation would be reduced.

Reasons for cost re-estimation	Abu Dhabi	Dubai
Informal re-estimates performed during development.	8%	13%
Formal re-estimates performed at pre-defined milestones/targets.	38%	6%
An amendment changed the system being built and a re-estimate required.	36%	4%
Variation order – Job/scope		13%
Re-estimation was not performed	18%	64%

Table 3.22: Cost re-estimates reasons

Table 3.23 shows the parameters that were included in the estimation process.

Responses	Abu Dhabi	Dubai
Software	18%	16%
Hardware	19%	11%
Number of delivered units (Scope of work)	9%	11%
Testing	12%	17%
Integration	14%	17%
Documentation	12%	17%
Training	14%	11%
Other	2%	--

Table 3.23: The items were included in the estimation

Table 3.24 shows the reasons for re-estimation. Responses were quite clear and indicated that budgetary approval, internal person-power requirement planning to gain client requirements and resource project planning are given similar attention. As indicated previously, both government departments in Abu Dhabi and Dubai are following the same federal rules and regulations and that is why the results are almost the same.

Purpose for which the effort estimate was done	Abu Dhabi	Dubai
Budgetary approval	36%	33%
Internal person-power requirement	29%	20%
Gain client requirement/automate the business process-re-engineering it/client change scope	21%	27%
Planning and project resource/allocation/upgrades	14%	20%

Table 3.24: Purpose of re-estimation

Table 3.25 shows that a large proportion of the IT team were involved in the (informal) estimation process.

Responses	Abu Dhabi	Dubai
Technical	6%	14%
C.E.O/President	--	4%
Database Manager	3%	7%
IT Consultant/Technical Advisor	13%	11%
IT Engineer	3%	7%
IT Manager/PM	29%	17%
System Analyst	6%	17%
General Manager	3%	4%
Development Manager	9%	7%
Programmer	6%	4%
IT Director/Manager	19%	4%
Strategic Planning/Project Admin.	3%	4%

Table 3.25: IT players involved in cost estimation

As shown in Table 3.26, the project manager, technical advisors and a large proportion of the IT team were involved in the (informal) estimation. It was noted that none of the technical teams or project managers involved in cost estimation have been trained in cost estimation. However, most of them appreciate the importance of using a well-established model for software cost estimation.

Project manager training	Abu Dhabi	Dubai
Project manager took training	40%	75%
Did not join training	60%	25%

Table 3.26: Project managers involved in advanced training

The interviewees highlighted the lack of proper training and awareness about cost estimation approaches. Hence it is strongly recommended that government employees who are involved in software cost estimation should attend formal training on cost estimation. In addition to that, a model should be built to enable project managers and technical employees to estimate the cost effort for software development projects. These conclusions result from both interview discussions and the questionnaires completed.

The interviewees indicated that proper planning is the most practical way to improve software estimation and that formal training on effort estimation is critically important. Other interviewees indicated that management commitment and historical analysis are important for cost estimation (see Table 3.27).

Recommendations to improve software estimation	Abu Dhabi	Dubai
Past experience of people involved	31%	13%
Management commitments and skills	23%	17%
Historical analysis of previous conducted projects	12%	13%
Proper planning	15%	31%
Formal training about effort estimation	19%	26%

Table 3.27: Recommendations to improve estimation

Despite the fact that 80% of the people involved in the cost estimation process had previous experience, (as shown in Table 3.28) there were still major discrepancies between estimations and actual efforts.

Previous experience in software effort estimation	Abu Dhabi	Dubai
Yes	75%	80%
No	25%	20%

Table 3.28: Experience in effort estimation

In addition to the above, it was noted that person-months and budget are among the most preferred parameters to determine the size of software projects. After that, functionality was the third highest preferred method (see Table 3.29).

Items organisation considered to determine projects size	Abu Dhabi	Dubai
Person-months	36%	22%
Budget	24%	21%
Functionality	11%	15%
Lines of code (LOC)	--	--
Function Points	3%	6%
Feature Points	--	9%
Number of bubbles on a data flow diagram (DFD)	5%	3%
Number of entities on entity relationship diagram (ERD)	8%	9%
Count of process / control boxes on a structure chart	5%	6%
Amount of documentation	3%	3%
Number of objects, attributes and services on an object diagram	5%	6%

Table 3.29: The variables that play a major role in defining the project size

A framework for measuring the size of projects is shown in Table 3.30. The framework was developed by the author after completing a process whereby the recipients themselves were required to identify which items were important for project size. Then the recipients were asked to evaluate each item. The first part of the process (essential item identification) took the form of a Yes/No answer to 12 items which have been identified as important (Futrell, Shafer, and Shafer, 2002). As well as identifying the essential items, the recipients were asked to relate the relative importance of each item to particular project size (e.g. a large project may impact on the organisation in terms of resources such as person-power to a greater extent than a small project).

Categories / items		Department in Abu Dhabi										
Measure the size of your projects		Org1	Org2	Org3	Org4	Org5	Org6	Org7	Org8	Org9	Org10	Org11
Person Moths	Experience Skill(Year)	2(5)	2(5)	3(2)	2(>5)	3(10)	3(>10)	2(5)	2(2)	3(15)	3(2)	2(15)
	Duration Month(level)	10(S)	6-12(S)	10(S)	48(S)	12-18(S)	3(S)	18-24(S)	27(L)	9(M)	6(L)	3(M)
Budget		L	L	M	M	L	V	S	V	M	V	L
Functionality		L	M			L	V			M		L
Lines of Code (LOC)												
Function Points (System Requirements)										M(100)		
Feature Points (User Requirements)										M(20)		
Number of bubbles (details) on a data flow diagram (DFD / UML)			L(25)				V(20)					
Number of entities on entity relationship diagram (ERD)			L(50)				M(50)			M(100)		L(200)
Count of process / control boxes on a structure chart								L (300)		M(25)		
Amount of documentation										M()		
Number of objects (classes), attributes and services (methods) on an object diagram				M(15)				L (300)				

Categories / items		Department in Dubai								
Measure the size of your projects		Org1	Org2	Org3	Org4	Org5	Org6	Org7	Org8	Org9
Person Moths	Experience Skill(Year)	3(5)	2(5)	1(2)	2(>5)	2(10)	2(>10)	2(5)	1,2,3 (2,5,10)	3(15)
	Duration Month(level)	6(S)	7(S)	6(S)	6(S)	12(S)	6(S)	6(S)	60(L)	12(M)
Budget		L	L	L	S	M	V	L	V	L
Functionality		P	F	F	F	P		F	F	P
Lines of Code (LOC)										
Function Points (System Requirements)		M(50)						L(100)	L(50)	S(20)
Feature Points (User Requirements)		P(20)		S (6)				L >200		
Number of bubbles (details) on a data flow diagram (DFD / UML)		(L ≥ 50)						(V≥50)		(L ≥ 25)
Number of entities on entity relationship diagram (ERD)		M(10)		L (25)			M(50)	V (200)	P(20)	M(10)
Count of process / control boxes on a structure chart							V(20)			
Amount of documentation								V (300)		
Number of objects (classes), attributes and services (methods) on an object diagram								V (300)	M(25)	

Experience: ❶ Junior ❷ senior ❸ Principal ❹ Distinguished
Levels: S – Small, M – Medium, L – Large, V- Very large
Budget: S – Small - $20k - $50k
M – Medium - $70k - $100K
L – Large - $200-$300k
V – Very Large >$300k
F – Full cover
P – Partially cover

Table 3.30: A framework for measuring size of projects

3.6.3 Conclusions of Survey 1

The survey indicated that there was no proper model used in the UAE for software cost estimation, which resulted in failures of most of the application development projects. The study recommended that a software cost estimation model should be constructed. The first step towards creating this model was determined. It was believed that identifying the parameters to determine the project size plays a big role in deriving the effort cost estimation model. It was noted that three variables were used in most government departments when determining the project size and these may be used in the method. The variables are person-power (person-days), budget and functionality. This, in turn, might help in determining the appropriate parameters for defining the applicable cost estimation model. It is apparent from the results that there is no clear definition of larger (versus smaller) project size.

Survey 1 has clearly shown that there are discrepancies between the estimated budget for software development projects and the actual budget. It is possible that this difference is due to the lack of an adequate cost estimation model, the absence of proper training, and the lack of experience gained from previous projects. Therefore, building a national database of previous projects, as well as defining a model that takes into account the project size and the skills involved, will all contribute considerably towards improving software effort estimation.

The main objective of Survey 1 was to analyse the key findings in the previously conducted studies of government departments in the UAE about the software cost estimation efforts. It also summarised and compared the most advanced techniques that are used, as well as determining the parameters that play a role in identifying the size of projects.

The software project data in IT departments were selected to determine the variables that play a major role in defining project size and type of application. In order to determine the approaches that government departments followed in estimating software development effort, the author started the study by inquiring about the approaches that the departments selected for developing their applications.

Various types of applications were examined. In addition, an investigation was carried out to examine whether most of these efforts were expended to develop fully functional software from scratch or to customise ready-made software, or for minor or major system upgrades.

The relationship between government departments and software vendors was also studied carefully. The main objective was to find out how much time government departments spend on software development. It was concluded that none of the software development estimation approaches used are accurate, and that all the projects studied were over budget.

In an effort to understand the strengths and weaknesses of the software development cost estimation approaches adopted, the author spent a considerable amount of time collecting data related to how the cost estimation process was conducted, in which phase it had been carried out and which skills had been utilised to carry out the process. In addition to the above, the main reasons behind the discrepancy between the estimated and actual effort were investigated.

3.7 Survey 2

The aim of Survey 2 was to determine the strategies that government departments use for managing software development. It also aimed to define the parameters that play a major role in defining project size. In an effort to determine a new, improved, cost estimation model, several parameters are proposed as a result. These parameters have not been considered in previous studies. The identified parameters have been categorised into seven groups; Organisation Type, Application Type, Organisation's Line of Business, Organisation Culture, Project Leadership, Project Technical Environment and Year of Project Completion.

Diverse readings on the importance of cultural and leadership characteristics were consulted including major anthropological encyclopaedias (Seymour-Smith, 1992).

The population of the UAE includes people from various ethnic groups and nationalities who work together in different fields and at different levels. The researcher's interaction with individuals from such backgrounds enhanced the understanding of the cultural environment. Moreover, invaluable advice was received from faculty working in different fields of social sciences and their input was considered in creating the survey. Finally, this project benefited a great deal from discussions and interactions from project managers working in the country.

Figure 3.1 shows the ratio of the respondents who completed the survey. As indicated (see Figure 3.1), 90% is the overall response rate from UAE organisations.

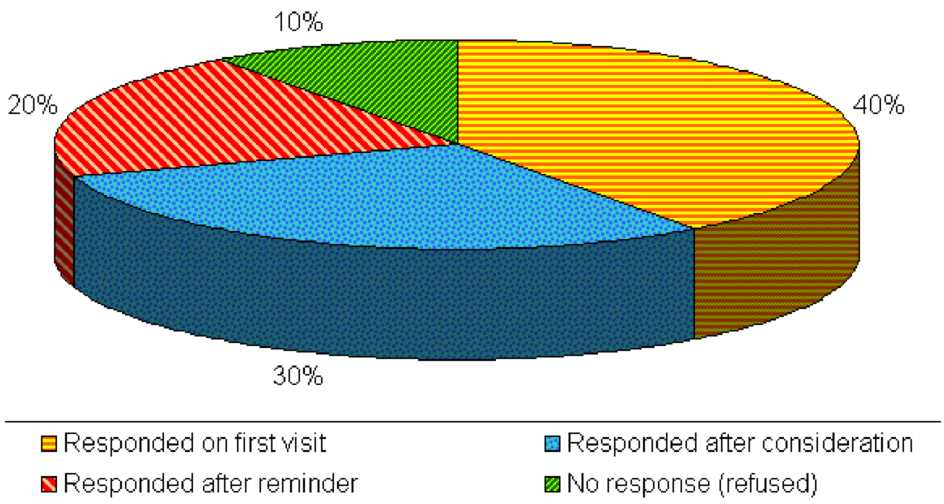


Figure 3.1: Overall response rate from UAE organisations

The departments completed most of the questions for the survey. The only data that was found difficult to capture was the number of business users and clients. The results showed that 75% of the questionnaires was completed. Few organisations felt that cost and effort were sensitive information which could not be divulged.

3.7.1 Organisation Type

The size of the organisation is based on the total number of employees. The organisations ranged in size from small to large-scale. Small scale organisations were defined as having between 40 and 200 employees, while large organisations had between 1200 and 16000 employees. Most organisations were either small or large, with a smaller number of medium-sized organisations (see Table 3.31).

Organisation size	
40 - 200	33%
1200 - 3000	42%
4000 - 9000	17%
10000 - 16000	8%

Table 3.31: Number of employees

The survey defined seven categories of 'Line of Business' (LoB). The majority of IT users were from government services departments and from the oil and gas sector (see Table 3.32).

Line of Business (LoB)	
Medical	2%
Governmental Services	41%
Telecommunications	2%
Public Services	10%
Tourism Services	5%
Education	2%
Oil & Gas or Energy	32%
Other	5%

Table 3.32: The ratio of projects in the LoB

The survey defined two application types (core and support). It was observed that 54% of the applications were core systems, and 46% of software development were support systems (see Table 3.33).

Application Type	
Core	54%
Support	46%

Table 3.33: The application type for projects within the organisations

Organisation type was defined as being one of three groups. 39% of the Organisations were 'Project Oriented', while 51% were 'Matrix structured', and 10% were 'Functional' (see Table 3.34).

Organisation Type	
Project Oriented	39%
Matrix	51%
Functional	10%

Table 3.34: Organisational types

3.7.2 Application Type

Application architecture had three categories: Stand alone, Two Tier, and Three Tier. Different projects use different application architectures. It was noted 61% had a Two Tier application architecture (see Table 3.35).

Architecture Type	
Stand alone	7%
Two Tier	61%
Three Tier	32%

Table 3.35: Application architecture

It was noted that more than 30% of the departments interviewed have average monthly transactions between 1,000 and 5,000, whereas, 24% have less than 1,000 transactions per month (see Table 3.36). A transaction is an action which reads and transforms the values of records and devices. A collection of actions which comprise a consistent transformation of the state may be grouped to form a transaction (Gray, 1981).

Number of Transactions (per month)	
< 1,000	24%
1,000 - 5,000	29%
5,000 - 10,000	17%
> 10,000	5%
N/A (no records)	24%

Table 3.36: Monthly transactions for organisations

It was shown that projects of 6 to 12 months duration were 39%, whereas 24% of projects had a duration of less than 6 months (see Table 3.37).

Projects Duration (in months)	
< 6	24%
6 - 12	39%
12 - 24	20%
24 - 36	7%
> 36	10%

Table 3.37: Duration of projects

Oracle (58%) and Microsoft (32%) were the main database tools used (see Table 3.38).

Database Management System (DBMS**)**	
MS Database	32%
Oracle DB	58%
IBM Database	10%

Table 3.38: Database Management Systems (DBMS)

Most of the departments surveyed used Oracle (42%) or Microsoft (34%) tools (see Table 3.39).

Software Tools	
Microsoft tools	34%
Oracle tools	42%
Sun Microsystems tools	17%
IBM tools	2%
Other tools	5%

Table 3.39: Software tools

The most common Operating Systems used were Windows (61%) and Linux (12%) (see Table 3.40).

Operating Systems (OS)	
Windows	61%
Unix	7%
Linux Redhat	12%
Sun Solaris	10%
QNX, Linux, Win	10%

Table 3.40: Operating systems

3.7.3 Skill Sets

The skills of the team members were included to determine similarities within the organisation's culture. Various skill sets were examined in an effort to determine project similarity. Skill types were evenly distributed across all sectors, but Web developers predominated at 26% (see Table 3.41). Skill sets and levels give a picture of the total staffing, cumulative costs and relative risks. These parameters were not analysed thoroughly since it was found the impact of leadership and culture were stronger.

Skills	
Application Manager	12%
Business Advisor	11%
Technical Advisor	19%
Programming System Analyst	19%
Web Development	26%
Project Manager	13%

Table 3.41: Skill type involved in a project

As regards staffing levels, it was found that most projects employed 5-10 people (see Table 3.42).

Total team size	
< 5	19.5%
5 - 10	56.1%
11 - 20	19.5%
21 - 30	4.9%

Table 3.42: Number of the team members

3.7.4 Application Complexity

One of the major issues in software cost estimation is the estimation of the size of a software development. General parameters have been considered when determining the cost of software development and when identifying similarity among projects. It was noted that the application's complexity was not often measured. Source lines of code (SLOC), Objects, Data Flow Diagrams (DFD) and Unified Modeling Language (UML) were not often used by organisations. Most projects used an Entity-Relationship Diagram (ERD), Forms and Reports (see Table 3.43).

Application Complexity (Entities)	
Line of Code (LOC)	9%
Process	16%
Objects / Classes Diagrams	7%
Tables/Entities	24%
Number of bubbles (DFD / UML)	3%
Reports / Forms:	29%
Function Points (FPs)	12%

Table 3.43: Measurement of application complexity

Table 3.44 identified the actual cost of IT projects. It was noticed 17% of projects cost over US $1 million. Some of the costs were not available for commercial confidentiality reasons.

Actual Cost in USD	
> 5.4 M	2%
1.37 M - 5.4 M	12%
556 K - 1.36 M	7%
146 K - 550 K	15%
60K - 145 K	17%
< 60K	10%
N/A (no records)	37%

Table 3.44: Real cost of projects (in USD)

Table 3.45 shows that 90 to 470 person-days represents 59% of the actual effort.

Actual Effort	
>1420	2%
1181-1420	10%
950 - 1180	5%
711- 949	7%
471 - 710	12%
240 - 470	32%
90 - 239	27%
N/A (no records)	5%

Table 3.45: Actual effort (person-days)

3.7.5 Culture and Leadership Characteristics

Leadership data was processed and presented as follows. The six characteristic attributes identified in Chapter 2 were:

- Interaction and relationship with team members
- Leadership decision-making
- Communication skills
- Ability to understand the project and organisation's culture
- Active thinking
- Ability to motivate team members

Each characteristic was divided into 4 further sub-items that were rated 1 to 9 by the respondent. Then for each of the six main characteristics the average of the sub-items rating was calculated. The leadership characteristics part of the questionnaire is shown in Appendix C.

Project team culture characteristics were similarly assessed, as shown in Appendix C. The main cultural characteristics as identified in Chapter 2 were:

- Timeliness - attitude toward time
- Collaboration
- Job stability
- Intercultural intelligence
- Team experience
- Communication
- Reward mechanism

The results are shown in Table 3.46.

Cultural Characteristics	Average of Respondents Rating (1-9)
Timely - attitude toward time	6.7
Collaboration - team relationship/impersonal relations	7.0
Job stability – reflection of loyalty	6.8
Intercultural intelligence	7.1
Team experience	6.6
Communications	7.2
Reward mechanism	6.5
Cultural average	6.8

Leadership Characteristics	Average of Respondents
Interaction and relationship with team members	7.0
Decision-making	6.9
Communication skills	7.2
Understanding of the project and organisation's culture	6.7
Active thinking	7.3
Ability to motivate team members	6.5
Leadership average	6.9

Table 3.46: Culture and Leadership characteristics

It is apparent that the recipients recognised that leadership and culture are important, because the average result was approximately 7. Indeed, it can be shown that there are strong correlations between leadership and cultural characteristics (see Chapter 4 for further detail).

3.7.6 Conclusion from Survey 2

As with the previous research (Hamdan et al., 2005), it is again recommended that a national database should be built. It should store historical data and lessons learned from previous government projects. This database would be of great help, as many of the software development projects in government departments are similar.

Despite the unique and distinguished leadership styles used in managing software development projects, the lack of historical data and models has impacted negatively upon the effort estimation process. The creation of a well-defined model that takes into account the size and the skills involved in a project and the flexibility of the rules and regulation, will help considerably in improving software effort estimation.

Most government departments used expert judgement based on previous experience as a tool for software cost estimation and there was no well-defined cost estimation model being followed. In addition, the estimation process happens once a year during the budgeting process, and IT management tends to inflate the number to be on the safe side. The software vendors, such as Microsoft, IBM, Oracle, etc., also use expert judgement in close co-operation with the software companies. The results from Survey 2 also highlighted the importance of leadership and culture in determining the cost estimation.

3.8 Summary

This chapter has described the survey development and data collection and analysis. A staged approach, in three phases was used, to finalise and consolidate the survey in order to include the role of leadership and culture in cost estimation.

The research so far has identified that there is no single formal estimation method or model in use for software cost estimation. From questionnaires completed by the various organisations in the UAE, it appears that failures and over-runs in most of the application development projects can be attributed to the lack of an estimation model. Therefore, it is recommended that a common software cost estimation model, which takes into account the culture of the countries, should be constructed for the UAE and the Gulf States (Hamdan, et al., 2005).

The parameters that determine the project size are person-power (person-days), budget and functionality. It is apparent from the results that there is no clear definition of large or small project size. Therefore, uniform interpretations of the semantics of a software project's size attributes are necessary to provide a guide to the activities by setting definitions for each attribute.

Various researchers have developed models for estimating software development effort (Shepperd and Schofield, 1997). Few have considered or focused on managerial issues (Boehm, 1981) and organisational culture and the leadership characteristics of project managers has not been assessed. This thesis investigates the hypothesis that organisational culture and project leadership play significant roles in determining the effort required for software development projects. In order to determine and validate the hypothesis, a survey of software development projects within government departments in the Gulf States has been undertaken, and the impact of organisational culture and leadership on software effort estimation determined.

CHAPTER 4

Software Project Measurement

4.1 INTRODUCTION

The domain of software project measurement is complex. The lack of a uniform measurement protocol exacerbates this complexity (Myrtveit et al., 2005). There is a need to develop techniques to describe and register knowledge of project resources. Here, it is proposed that the use of ontologies will help in obtaining a common understanding in measurement protocol.

In this chapter, a framework for software project measurement is presented. This framework is used to measure the attributes highlighted in Chapter 3 and to assist in the statistical analysis developed in Appendix D. The idea behind this framework is to present a set of guidelines which capture the definitions and relationships for each attribute used in measuring cultural and leadership factors, to ensure that different data are analysed accurately. These attributes are included in the ontology system and Profile Theory. Ontology and Profile Theory provide ways of interpreting the factors. Thus the same measure can mean different things to people from different cultures, even when the same factors are being considered, resulting in misunderstandings and conflicts. The use of an ontology or Profile Theory is intended to address these issues.

The rest of the chapter is organised as follows. Section 4.2 describes the ontology-based framework. Section 4.3 presents the software effort estimation ontology system and objectives. Section 4.4 gives an overview of the profile theory approach as a basis for a general theory to assess leadership capabilities and compatibility. Section 4.5 and 4.6 provide selections of measurement techniques that are assigned to attributes and entities to avoid measurement problems. Section 4.7 identifies units, scales and values to measure attributes. Section 4.8 provides a summary of the chapter.

Before elaborating on the cost estimation model, it is essential to identify the major parameters. A survey (described in Chapter 3) was used to measure the perceived impact of the culture and leadership factors on software development projects. These characteristics were reviewed to ensure that a comprehensive list of measures was included in the cost estimation model.

Software cost estimation is an essential activity throughout the software life cycle. Cost estimation may be performed before the project initiation, during software development, or after the project is complete. After completion of the project, actual values are compared with the estimates to determine the accuracy of the estimation. Good estimates are useful for project productivity assessment, for initial validation, for monitoring the project's progress and for deciding whether the project ought to proceed. Inherent difficulties, such as accuracy of estimates, data availability, number of parameters being measured, all complicate the choice of sound assumptions. Moreover, there is a lack of a consistent interpretation of project attributes within an organisation and across organisations. Thus, improving project cost estimation is a necessary endeavour in software development and one which encompasses two major fronts: modelling and measurement.

There has been a continuous search for better tools to aid project managers in the measurement process (Fenton and Pfleeger, 1997; Kitchenham et al., 1999; Briand et al., 1999). Apart from effective measurement tools, protocols are also necessary. A measurement protocol is a formal description of how an attribute is measured. Attributes and measurements should be defined unambiguously. Without proper definitions and clear guidelines, it is difficult to derive meaningful comparisons or to ensure repeatability of measurements (Kitchenham et al., 1995). Moreover, any measurement methodology that is proposed must take into account the fact that project managers are unlikely to apply techniques that are difficult to understand.

Jørgensen and Shepperd (2007) identified over 300 papers on software cost estimation. These deal mostly with uncertainty assessments and code complexity. However, cultural and leadership aspects were not addressed explicitly. Boehm (1981) and Krishnan (1998) discussed ways to improve cost models and project management (i.e., capability of a software team) and argued that these are the most significant factors in determining cost or quality in a software product. Many attempts have also been made to identify the effect of individual differences in software developers (Krishnan, 1998), but these neglect leadership and cultural issues.

Another major challenge is deciding how data items can be categorised and mapped to an appropriate form that can be used in a cost estimation model. Techniques such as ontologies (Gruber, 1993), System Dynamics (Forrester, 1968; Abdel-Hamid and Madnick, 1991), and Profile Theory (Plekhanova, 2000) have all been applied to overcome this problem. In the following sections, an assessment of ontologies and profile theory is elaborated to determine how to adapt them within the proposed model.

4.2 Ontology-Based Frameworks

Gruber (Gruber, 1993) defines ontology as a means of formally specifying a common understanding of a domain of interest in terms of concepts, relationships or axioms. This definition gives an explicit representation of knowledge and shared conceptualisation of a domain that is commonly agreed to by all parties (e.g. for constructing detailed explanations for users). Ontologies give frameworks to problems that are not obvious by identifying profiles of attributes and their relationships. Research on ontologies has become increasingly important across a wide range of sciences. Ontologies have been used in many domains (Sheth and Lytras, 2007) with application in fields such as e-commerce, information retrieval, artificial intelligence, software engineering, and biomedical informatics.

Bergmann and Schaaf (2003) identify the use of ontologies as complementary to CBR. The vocabularies (variables) used in the two approaches are basically similar. They both use:

- formal modelling for restricting the possible interpretations of knowledge items (e.g. documents, processes),
- semantic-based access to knowledge items, and
- similar representational paradigms.

Both approaches require some formalisation in advance to represent knowledge and use a frame-like representation to record knowledge. Ontologies are richer in terms of the structural modelling primitives and can be used to display aspects of the domain in terms of concepts and properties as well as to show how these are related to each other. They typically use logic-based inferencing techniques such as frame-logic (f-logic) or description logic (which is related to first-order logic) to reason about these structures with their consequent benefits and limitations. These can be used to generate deductions or to infer facts not explicitly presented. CBR also uses a frame-based paradigm but then typically carries out similar computations to relate a new case to existing cases in the knowledge base. Typically, CBR approaches are less rich in their modelling primitives and use different techniques to liken new knowledge to existing knowledge, usually with the purpose of identifying existing solutions that can be applied to new problems. They are, however, more flexible in being able to relate items on a continuous scale rather than on a discrete scale. Ontologies (utility-based reasoning) normally require further processing. CBR retrieval can produce results, even in the case where no close matches exist (Myrtveit et al., 2005).

Ontologies can help in classifying new information such as a completed profile in terms of historic profiles already in the knowledge base. CBR can operate across multiple case bases, while ontologies can operate across multiple repositories using a single ontology or a family of ontologies. However, problems can arise when different representations are used in different locations, leading to potentially ambiguous interpretations. This can be avoided if standardised or shared representation languages are used (Bergmann and Schaaf, 2003).

Despite their differences, ontologies and CBR are to a point complementary. The more expressive representational framework of ontologies can be used to create richer domain models. These models are separate from the reasoning that may be performed on them. The traditional frame-logic, description logics, or CBR techniques such as similarity calculations and utility-based reasoning can be used to derive conclusions from the knowledge base or perform modifications on it.

An ontology can be developed to support a consistent measurement strategy. Sometimes two systems use the same terms, but mean different or contradictory things. As a result, when several people are asked to define the size of a project or the necessary skills for a particular task, or how long it takes to make a decision about a certain issue, they will often give widely differing answers. Ontologies can be derived to help highlight these differences and can be viewed as foundational and as structured as a dictionary. In this work, an ontology to support IT project cost estimation is needed to provide consistent measurement practices. This ontology would establish a common measurement protocol and provide a uniform interpretation of project parameters and their semantics.

4.3 Software Effort Estimation Ontology System (SEEOS)

An ontology defines the semantics of the project parameters and the relationships between them; it establishes shared concepts between different projects. Despite the usefulness of existing cost estimation methods, the lack of a shared ontology to guide the cost estimation protocols promotes inconsistencies in efforts and results. To address this issue, an 'ontology of software development' has been developed by the author in order to categorise the semantics of software development according to project factors. The uniform interpretation of these semantics provides a sound basis for guiding the application manager's activities by using a set of definitions for each attribute. It enables project managers to access a software project's features more accurately and consistently. Figure 4.1 shows the developed ontology. Following considerable consultation and interviewing, this ontology passed rigorous validation by numerous IT leaders in the industry and sociologists in academia. This would help in typology consensus agreement over main terms and concept definitions used in sociological perspectives.

With each project type, this ontology associates an organisation's line of business as a child. Each SEEOS node is associated with a set of entities and attributes associated with parameters and values. For example, the organisation's line of business values are medical, government services, telecommunications, tourism services, public services, education, and oil-gas. A system is categorised as either a support system or a core system. For example, each system has an IT budget that covers services, hardware/software cost, effort and training. The ontology classifies a junior as someone who has more than one year of experience but less than five years. A senior is identified as someone who has more than five years of experience but less than ten years. A principal is someone who has more than ten years of experience. Training is online and onsite. The service/support is classified as platinum, gold and silver.

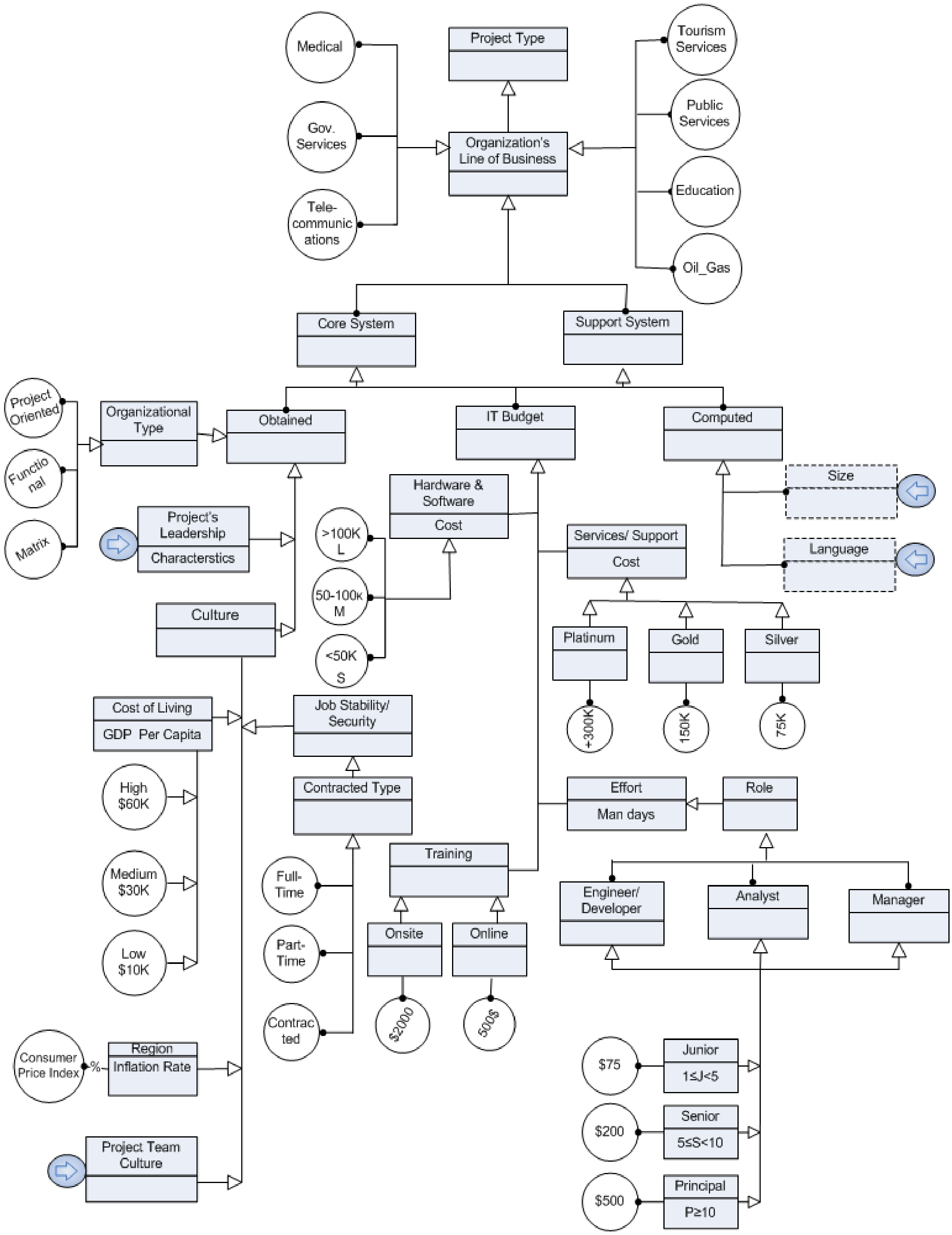


Figure 4.1: Software Effort Estimation Ontology System (SEEOS)

The computed ontology covers programming languages and project size (see Figure 4.1). These two nodes are elaborated in Figure 4.2 and Figure 4.3 respectively. The obtained ontology includes organisational type, project leadership and project team culture characteristics. Culture is identified as cost of living, job stability and security, geographical region, inflation rates and project team culture. The ontology classifies living expenses as high, medium or low gross domestic product (GDP) per capita. Job stability is either full or part-time. The languages ontology associates each language to second, third, or fourth generation (see Figure 4.2).

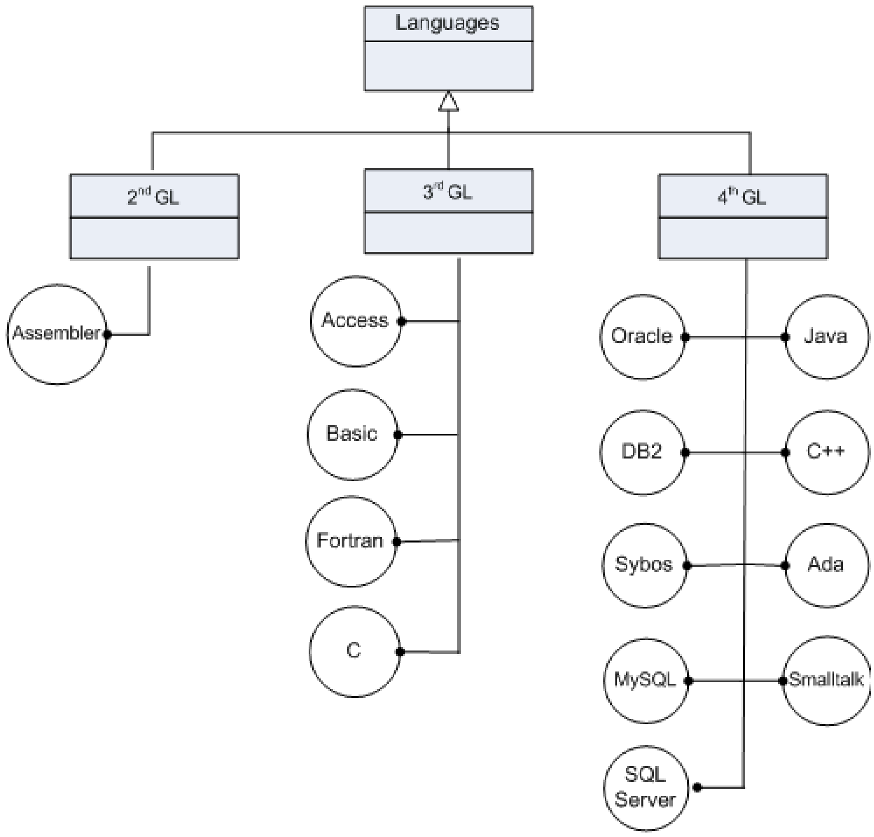


Figure 4.2: Programming languages ontology

The project size ontology (see Figure 4.3) measures six different values. This shows the ontology of different project sizes in terms of function points, source lines of code (SLOC), data flow diagrams (DFD) or unified modelling languages (UML), entity relationship diagrams (ERD), classes or objects and processes and structure charts. The function points computed by unadjusted function points (UFP) and value adjust function points (VAF = 0.65 + ((sum of all General

System Characteristics (GSC) factors) / 100)). The values of the corresponding project size are specified as simple, medium or large.

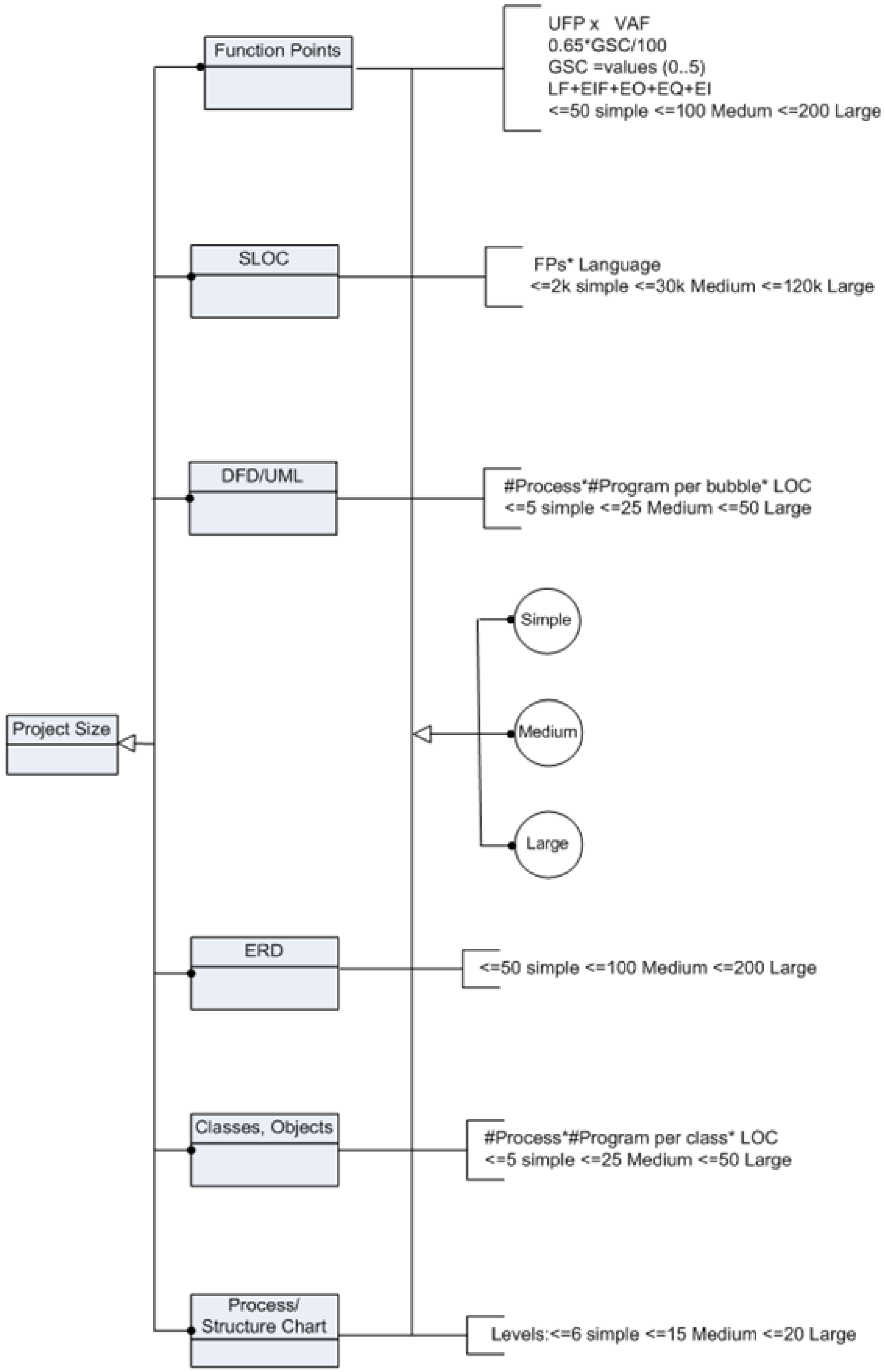


Figure 4.3: The project size ontology

The culture ontology is shown in Figure 4.4.

- Timeliness refers to attitudes towards time. This means the ability to manage time and available resources effectively in a way that does not detract from regular schedules.

- Collaboration measures the work balance between personal and professional roles. Success depends on the whole team.

- Job stability includes job security and refers to the importance of the working relationship between the leader and team members and the mutual trust, respect and cooperation that this requires.

- Intercultural intelligence refers to the ability to understand other cultures' world views and also implies an understanding of interpersonal relations, which involves the consideration of other people's feelings, values and goals.

- Rewards refers to the giving of rewards and offer of incentives by the leader who thus encourages and supports the team's professional development and achievements.

- Communications means that the leader supports the open exchange of ideas and encourages team members to become leaders themselves by using their own decision-making capabilities in project development.

- Team experience refers to the relevant skills and knowledge that team members possess with regard to particular projects.

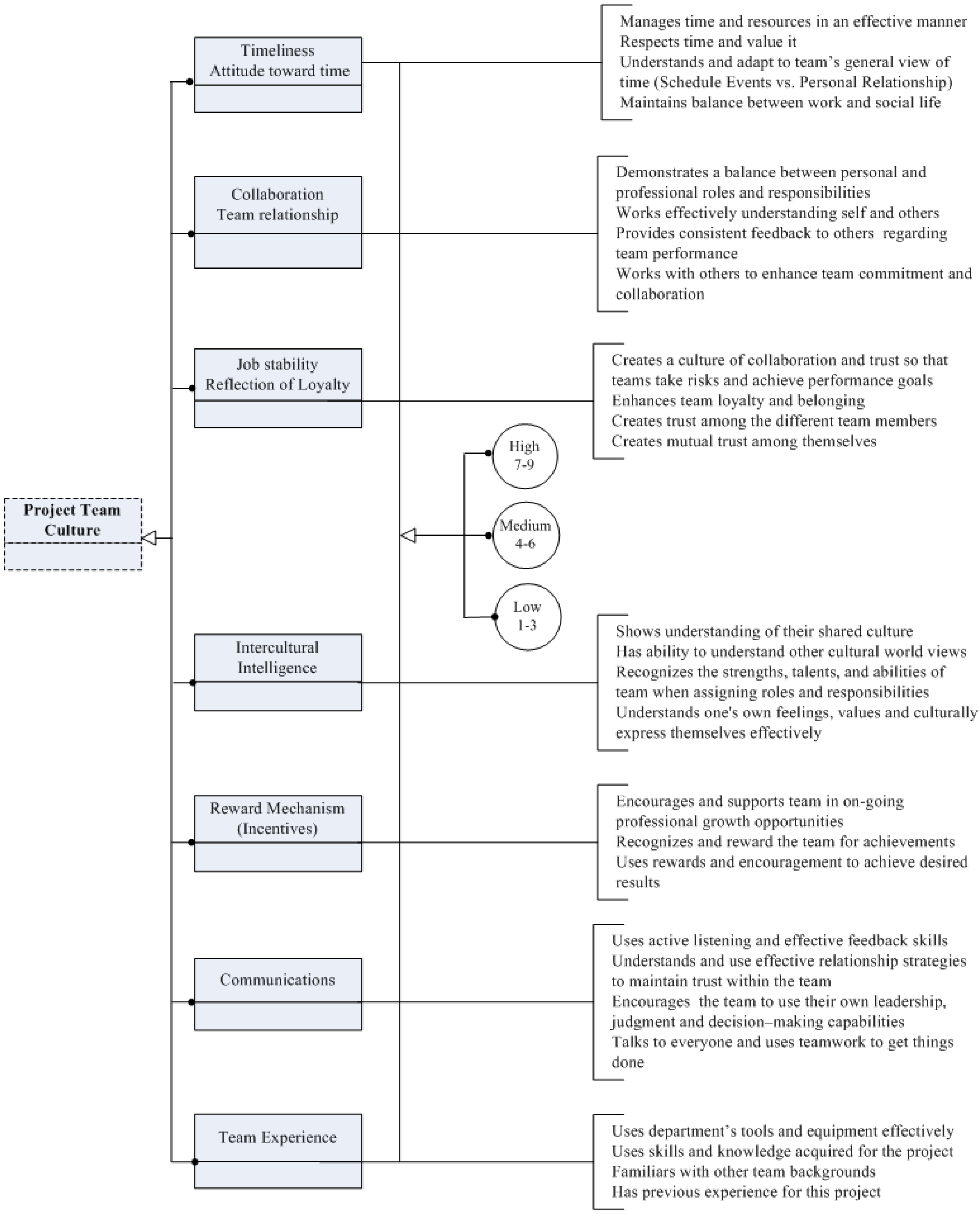


Figure 4.4: Culture ontology

The leadership ontology categorises several attributes (see Figure 4.5).

- Interaction behaviour and relationships with team members means that the leader creates learning experiences, appreciates completed work and treats team members with respect.

- Decision-making in leadership means, optimally, that the leader makes the right decisions in a timely fashion and consults with the team about the organisation's direction.

- The ability to motivate means that the leader shares goals and gives appropriate instructions, support and encouragement to team members.

- The ability to understand project leadership and the organisation's culture means that the leader is able to understand and manage multicultural teams. Leaders and teams need to recognise that all team members are equal and that they all belong in the organisation.

- Active thinking means that the leader enhances team contributions and sets feasible targets.

- Communication skills means that the leader communicates with team members effectively and vice versa. Some of these skills include being active participants by listening to others attentively and giving appropriate and positive feedback.

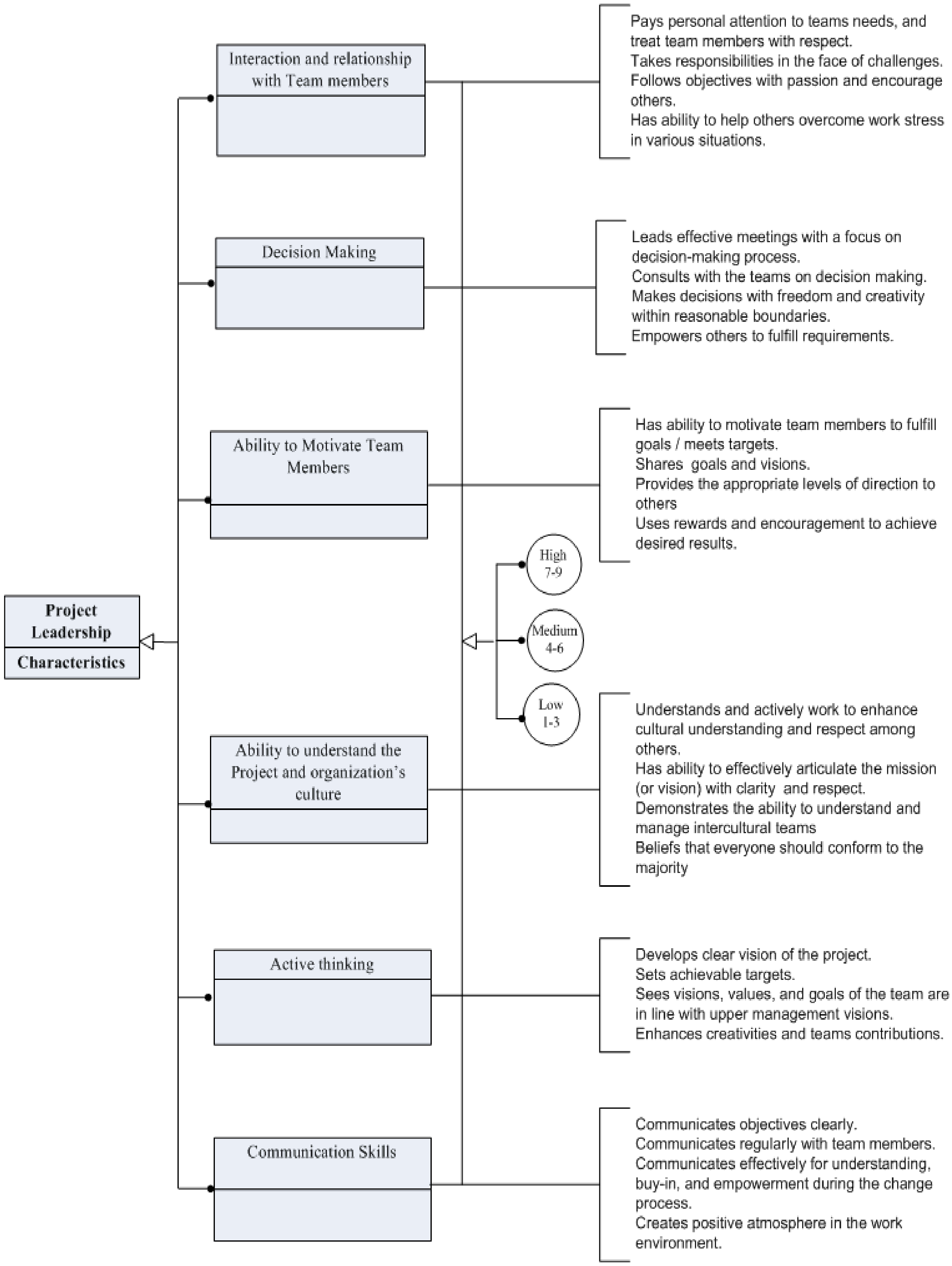


Figure 4.5: Leadership ontology

4.4 A Profile Theory-Based Model of Leadership

Combining leadership characteristics with other variables is difficult and involves quantitative measures of capability. The completeness property is important for the identification of the essential profile factors and it is equally important for these to be incorporated into the profile description. One issue that still needs to be addressed is how to use profile theory to describe leadership. The use of profile theory technique was addressed in Appendix F.

4.5 Identifying the Key Cultural and Leadership Factors

In Figure 4.6, the diagram shows factors affecting team culture and leadership. The first feedback loop starts at the node labelled "Leader" and follows the green arrows. The second one starts at the node labelled "Team Culture" and follows the blue arrows. There are two main positive feedback loops. For example, good team culture improves punctuality and timeliness which, in turn, reduce effort and cost. Clear estimates and successful projects will improve reward systems, resulting in job stability. Also, as team members' subjectivity is avoided, collaboration will increase. This feeds back into the team culture which completes the loop.

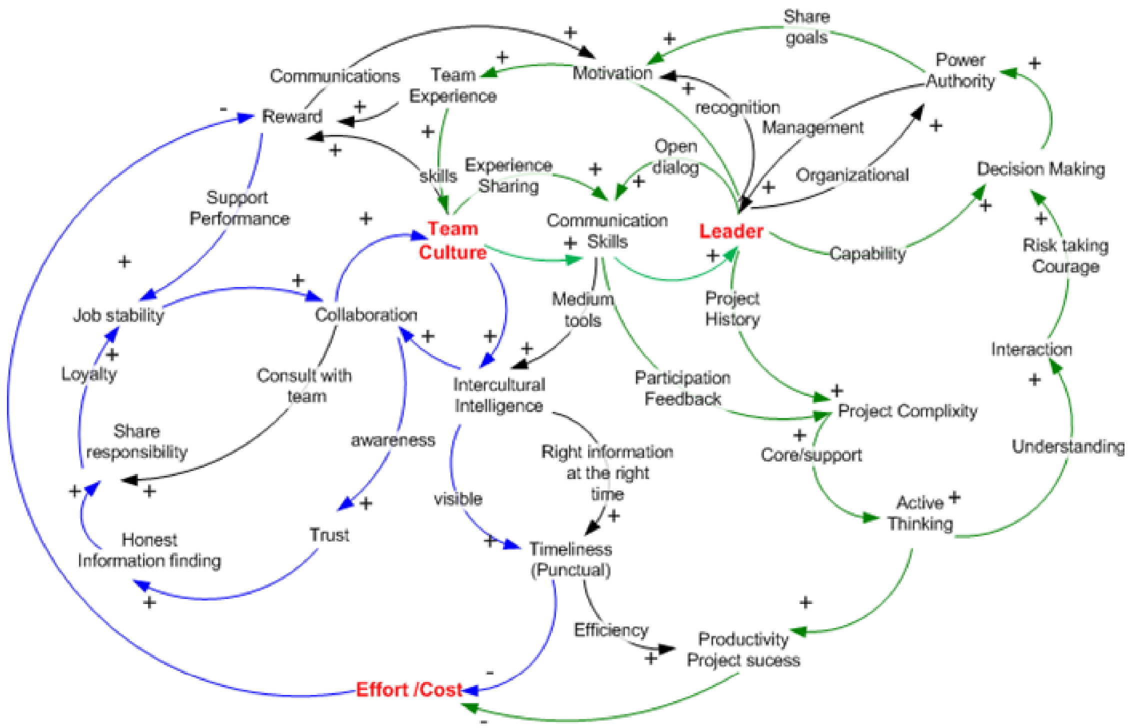


Figure 4.6: Factors and their relationships affect team culture and leadership

The other feedback loop represents good leadership based on recall ability of historical data and active thinking. This increases productivity, resulting in lower cost and effort which, in turn, gives better rewards and support to the team. Leadership increases collaboration and team culture through sharing experiences and communication skills. Again this feeds back into leadership and completes the loop.

An important way to analyse a team or its leader is to study their characteristics through quantitative approaches. For example, what is more important - having an "excellent" or experienced team leader or having a perfect working environment? Which has more impact? It is important to define the common characteristics which contribute to a software project's success and to list these characteristics as factors or dimensions (Kaner and Bond, 2004).

The selected factors were obtained from a checklist used in interviews and are those that most leaders and team members consider to be important. In fact, personal characteristics relevant to the project have a great influence on work and project resource (Plekhanova, 2000). Based on observation, it was noted that the attributes interact in a complex manner. An effective strategy to understand these interactions is to visualise them as shown in subsequent figures (i.e. Figure 4.6 and 4.7). Figure 4.6 shows a causal diagram where leadership and cultural factors are influenced by a number of attributes. This diagram is based on the survey results and interviews. The derivation of the causal diagram was drawn from the correlation analysis (see Appendix D). In Chapter 6, a further examination provides an analysis of leadership and team culture and shows that these factors are significant in software development and cost estimation.

The leadership factors tree (see Figure 4.7) is derived from the feedback of factors and their relationships, which affect team culture and leadership. This tree represents the factors that are considered important as seen in the generic causal diagram (see Figure 4.6). The leadership factors are classified as a set of characteristics. The factors are each given weights, based on their importance. The common factors that define leadership are:

- The Leader (the attributes which define all characteristics of the leader)
- Team culture (capabilities, knowledge, experience and skills)
- Organisation type
- Communication skills
- Project complexity

Weight preference = important or not important.

The values of leadership attributes were measured independently. These measures help to explain job specifications and enhance the decision-making process in finding a good match between projects and project managers. The decision to assign a particular project manager to a certain project or to hire a project manager requires that the capabilities and compatibilities of the manager be assessed in a quantifiable manner. Therefore, it is important to identify attributes that define project managers (see Figure 4.7).

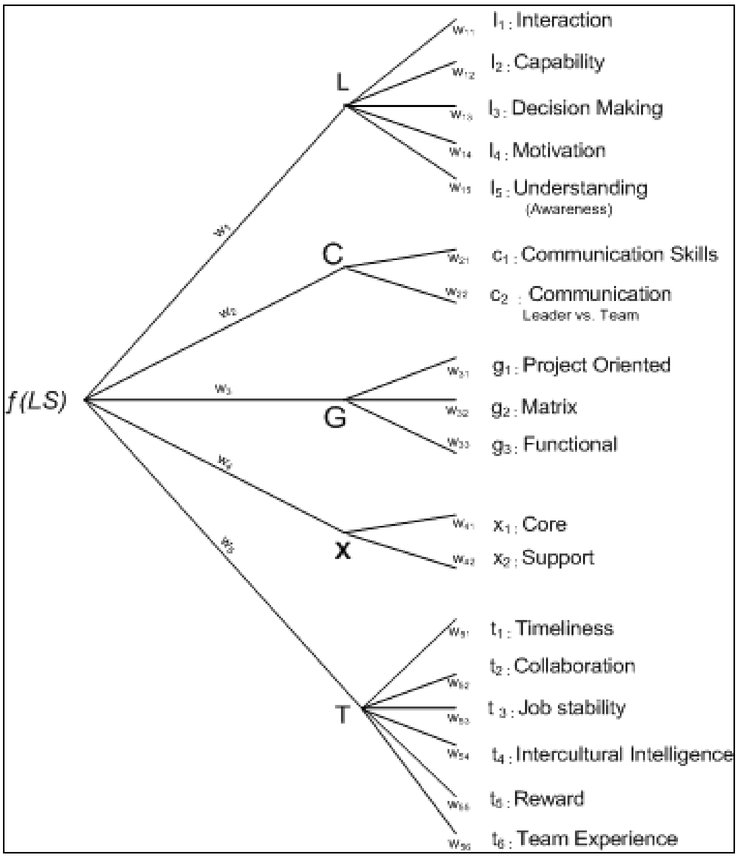


Figure 4.7: Leadership factors tree

Combining leadership characteristics with other variables is difficult and involves using quantitative measures of capability.

The leadership factors tree was then introduced to indicate the weight of each factor and was also used for the notation of profile description. The tree, in its entirety, is important for the identification of the essential profile factors and for these to be incorporated into the profile description. Thus, it is possible, using this notation of profile presented in Profile Theory, to describe leadership. For example, factors that represent leadership (LS) could be described as follows:

$$f(LS) = \{<\varepsilon_1, L, \omega_1)>, <\varepsilon_2, T, \omega_2>, <\varepsilon_3, G, \omega_3>, <\varepsilon_4, C, \omega_4>, <\varepsilon_5, X, \omega_5>\}$$

Where:

L : Leadership characteristics, such as style, power, capability, traits, and skills

T : Team characteristics, such as culture, knowledge, personal competencies

G : Organisational type, such as project-oriented, functional, or matrix authority

C : Communication skills, in both channels (leader vs. team culture)

X : Project complexity, such as core or support systems

ε_i: Factor existence, such as $\varepsilon = 1$, non existence $\varepsilon = 0$; where ε_i is

ε_1, ε_2, ε_3, ε_4, and ε_5 show the factor existence for leader characteristics, team culture, organisational type, communication skills and complexity, respectively.

ω_i: shows the total weight of sub-factor(s) - weight is divided equally in approximation ad hoc cases or based on importance or priority; ω_1, ω_2, ω_3, ω_4, and ω_5, are the weights for leadership characteristics, team culture, organisational type, communication skills and complexity, respectively. Further details are provided to illustrate the modelling of leadership factors using Profile Theory. This theory is an alternative approach (see Appendix E). It is not considered in this research except for an explanation of how it may be elaborated.

4.6 Selections of Measurement Techniques

This section will identify some possible methods of defining data sets used as a standard. As discussed earlier, ontology acts as a common dictionary to reduce semantic inconsistencies which arise when identifying attributes. This section discusses the definition of entities, value representations and identification of levels of entities in order to provide a standardised method.

There are several definitions of measurement, such as that of Fenton and Pfleeger (1997): "Measurement is the process by which numbers or symbols are assigned to attributes of entities in the real world in such a way as to characterise them according to clearly defined rules such as characteristic, attribute, trait, object, and situation" (Moses et al., 2002). Consequently, a measure is used to describe or characterise the attribute. Formally, measurement is identified as a mapping from the empirical world to the formal, relational world in such a way as to describe it (Fenton and Pfleeger, 1997).

Measurement is both fundamental and essential in empirical software engineering disciplines (Kitchenham et al., 1995). Several structures are highlighted:

- attribute: whether it actually exhibits the entity measured,
- unit: whether an appropriate means of measuring the attribute is used,
- instrument: whether any model underlying a measuring instrument is valid and the measuring instrument is properly calibrated,
- protocol: whether an acceptable measurement protocol is adopted (Kitchenham et al., 1995).

Various cost estimation models (Shepperd and Schofield, 1997; Kitchenham et al., 2002) are faced with similarly diverse interpretations of the measurement parameters. For example, similar project attributes may be categorised differently even when they use a similar unit of measurement (i.e., project size: function points or budget). It is important to reach agreement about definitions of attributes and their values. Measurement theory is an observation (act of assigning a value) that reduces an uncertainty expressed as a quantity. It includes factors such as assessment of attitudes, values and perception in surveys or aptitudes testing of individuals.

Most cost estimation methods require information about how past projects have been implemented. There have been attempts to set up an industry-wide database of past projects (ISBSG, 2003).

Measurement plays an important role in people's lives, and it is hard to imagine how things can be done without measures being taken before and after. Beside numeric measurements, questionnaires are used to make appropriate choices. In small software projects, measurement is essential to define the size of a software project and this is what makes a project successful. Here are some of the major general measurement-related issues:

- measurement of the size of software project: budget, functionality or effort
- large size budget project: 500,000, 1 million, 10 million
- optimum skills 'senior' (years of experience) for project manager: 5, 10 or 15
- good team leader: behaviour, skills or power of authority

4.7 Entities and Attributes

Entities can be defined as objects or events which can be observed (Kitchenham et al., 1995). These include products created, and processes used, in the development of software. The attributes are properties or characteristics of entities that are often described in a mathematical notation. Each entity has attributes or properties such as the size or cost of projects or skill types. For example, in many software projects, organisations measure in terms of number of lines of code (Fenton and Pfleeger, 1997; Park, 1992). Another study (Hamdan et al., 2005) revealed that organisations measured software projects by budget and functionality as a large scale or small scale project. The existence of measurement in any unit is important as long it can be conveyed in real world terms. It is also essential to identify attributes and entities before measurement is made (Moses, 1997).

Table 4.1 describes and summarises the software project attributes which were observed in selected organisations. The meaning of each question (survey in Chapter 3) was explained to each of the respondents at the interview, for which either Arabic or English was used, as appropriate. The attribute list for the project attribute entity is the 'Attribute' name, 'Description' and 'Type' of measurement. The attributes discussed were effort, organisation’s line of business, application type, organisational type, technical environment, organisation culture, leadership and year of project completion.

For example, numbers of entities (e.g. project size) were also discussed. These were sufficient enough to allow measurement of the project size property, such as line of code, function points, number objects (i.e. class diagrams), number of entities (i.e. tables, forms, reports), number of bubbles (i.e. DFD, UML) and number of processes (Futrell, Shafer and Shafer, 2002).

Attribute	Description	Type
Effort	Total Project effort	Measure
Organisation's Line of Business (Business Type)	The business types of Government and Semi-Government are Education, Health, Military, Civil Services, Oil, Tourism Services, Governmental Services and Telecommunications.	Measure
Application Type	The Systems are categorised into two categories: Supporting and Core Systems.	Measure
Organisational Type (system management type)	The project management authority and privileges. Each Organisation has assigned of System Management Type: Projectirised (Project Oriented) Functional (low Authority) Matrix (Intermediate Authority)	Measure
Project's Technical Environment	Number of Core Users (Backend Users, Number of Clients, Number of Transactions, Numbers of Entities, and Technology Used "Hardware and Software Infrastructure".)	Measure
Organisation Culture	The organisational culture highly impacts the process of cost estimation. Those are: • Timeliness • Collaboration • Job stability • Intercultural intelligence • Team experience • Communication • Reward mechanism	Measure
Leadership Type	The leadership characteristics impact software efforts and cost estimation. Those are: • Interaction and relationships • Decision-making • Ability to motivate • Understand organisational culture • Active thinking • Communication skills	Measure
Year of Project Completion	Most recent projects up to 5 years	Date

Table 4.1: Project attributes

Leadership and cultural characteristics were measured using multiple features. Leadership was measured by six items and each item was assessed by four sub-items. Cultural characteristics were measured by seven items and each item was also assessed by four sub-items. Communication skills were measured using two items combined with leader and team skill communications and each item was assessed by four sub-items. The cultural characteristics, leadership and communication skills sub-items were then measured using an ordinal scale with 9 points. The respondents were told to assign 1 to the lowest value and 9 to the highest value. The scores of the sub-items were then averaged separately for each item, resulting in a single item score with real values in the ranging from 1-9 (see Figure 4.8).

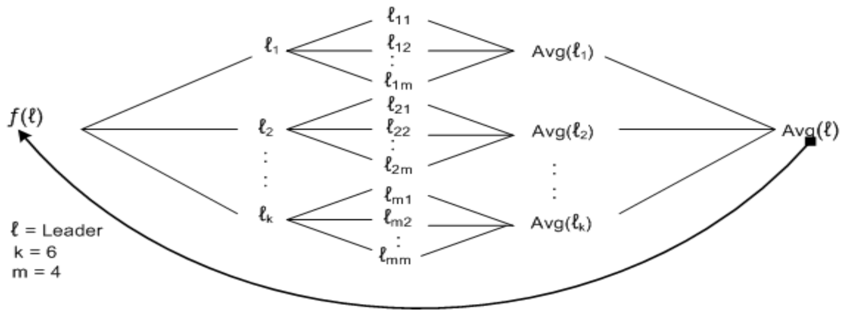


Figure 4.8: Averaged leadership characteristics

The 9-point scale was chosen to allow a broader analysis of the response. The scale ranged from exceptionally low to exceptionally high importance and was given a wider range of measurement and analysis. A scale range gives clear selection due to the importance of the attributes and sensitivity (Ahire and O'Shaughnessy, 1998). This also gives the responses a true value of their significance. The ordinal scale variables were treated as if they were measured using interval scales. This is reasonable, as shown by Briand et al. (1999), which stated that the ordinal scale variables are usable as intervals in the context of the analysis. The scale questions accommodate all points of view; the responses are coded to represent the average answer. In addition to the amount of effort (as numeric value), the dataset comprised the following factors:

- Organisation's line of business (range of value 1-7)
- Application type (range of value 1-2)

- Organisational type (range of value 1-3)
- Project's technical environment (numeric or string)
- Year of project completion (duration of project start and end date)
- Project's leadership characteristics
- Project team culture

The last two factors use a broad 9 point scale (exceptionally high = 9 points, very high = 8 points, high = 7 points, fairly high = 6 points, sufficient = 5 points, average = 4 points, low = 3 points, unsatisfactory = 2 points, exceptionally low = 1 points).

4.8 Units, Scales, Values

A measurement unit determines how an attribute is measured and is often associated with a scale. Attributes can be measured using more than one unit such as line of code or function points and can also measure different aspects of program size. The same unit can be used to measure different attributes. A unit must have a scale type. The scale type defines the mathematical transformation that can be made. Naturally, data vary in quality and the level of uncertainty associated with that variation may reflect on the effects of the analytical procedure. Furthermore, the reliability of data may differ based on a number of factors. As data are combined and transformed, the uncertainties may become narrower (Myrtveit et al., 2005).

The data convert into differently modified representations and transformations by deriving additional data that may offer insights for an analyst (Kitchenham et al., 1995). For example, dichotomisation of leadership and culture values into "low" and "high" allowed for a clearer analysis in discriminating between projects. It is also important that the transformation of the original data keeps all uncertainty attributes that influence the confidence assessments and decisions. There are five well-known scale types: nominal, ordinal, interval, ratio and absolute (count). The scale type explains the kind of operations it can perform. Table 4.2 presents a summary of the project unit.

Factors	**Description**	**Scale**
Effort	Person days (man days)	ratio
Organisation's Line of Business (Business Type)	Scale used to identify which Business type used (Education, Health, Military, Civil Services, Oil, Tourism Services, Governmental Services and Communications)	nominal
Application Type	The Systems are categorised into two categories: Supporting and Core Systems.	nominal
Organisational Type (system management type)	Scale used to identify which System Management Type used; Projectrized (Project Oriented), Functional (low Authority), Matrix (Intermediate Authority)	ordinal
Project's Technical Environment	Number of Core Users (Backend Users) Number of Clients, Number of Transactions, Numbers of Entities, Technology Used: (Hardware and Software Infrastructure)	absolute
Organisation Culture	Scale used to assess the impacts of process of cost estimation event • Timeliness • Collaboration • Job stability • Intercultural intelligence • Team experience • Communication • Reward mechanism	ordinal 1-9
Leadership Type	Scale used to assess the impacts of process of cost estimation events • Interaction and relationships • Decision-making • Ability to motivate • Understand organisational culture • Active thinking • Communication skills	ordinal 1-9
Year of Project Completion	Most recent projects up to 5 years	date

Table 4.2: Project unit summary

The attribute list for the project unit contains factors, description and scale types that identify values and scales of measurement for the attributes. For example, effort is measured in person days and culture and leadership use an ordinal scale, which measures the importance of these characteristics.

A reliability and validity analysis has been performed in this research. Reliability refers to the repetition of a measurement, whereas validity refers to the agreement between the value of a measurement and its true value (Myrtveit et al., 2005). The reliability of the multiple item measurement scale for the attributes was evaluated by using internal-consistency analysis (Cronbach, 1951). A coefficient alpha of 0.7 or higher is deemed satisfactory. Principal Component Analysis (PCA) was performed and reported for all attributes to provide a measure of convergent validity. The components with Eigen value greater than one were retained. Therefore, the convergent validity can be regarded as sufficient. Table 4.3 reports the reliability and validity analysis for the cultural and leadership attributes.

Leadership characteristics	items	Cronbach's Alpha	PCA
Interaction and relationships	1-4	.820	2.67
Decision-making	1-4	.905	3.16
Ability to motivate	1-4	.896	3.08
Understand organisational culture	1-4	.929	3.34
Active thinking	1-4	.940	3.40
Communication skills	1-4	.897	3.09
Cultural characteristics	**items**	**Cronbach's Alpha**	**PCA**
Timeliness	1-4	.955	3.54
Collaboration	1-4	.960	3.58
Job stability	1-4	.959	3.58
Intercultural intelligence	1-4	.891	3.04
Team experience	1-4	.728	2.36
Communication	1-4	.914	3.19
Reward mechanism	1-4	.909	2.54

Table 4.3: Coefficient alpha and principal component analysis of variables

The organisation’s line of business was measured using a seven-point scale: Medical, Governmental Services, Communication, Public Services, Tourism Services, Education and Oil and Gas. Application type was measured on a two-point scale

- Core systems and
- Support systems.

Organisational type was measured on a three-point scale

- Project-oriented (project manager has the highest power in making decisions),
- Matrix (project manager has moderate power in making decisions),
- Functional (project manager has the lowest level of power in making decisions).

Table 4.4 shows the scale range of each attribute range type, description and the level order. These attributes are then placed in the appropriate category, and the scale range entity will allow the ordinal or nominal scale to be refined within this context. Tables 4.4a - 4.4c identify all levels of culture value. The average across the team was used to determine the levels of culture value, as "low", "medium" or "high".

Range Type	Description	Level
Strong (high)	Project-oriented: has high project management authority	3
Moderate (nominal)	Matrix: project management authority has intermediate authority	2
Weak (low)	Functional: project management authority has low authority	1
High	**1. Timeliness**: team members respect each other's time and are keen to meet deadlines. This means that the team arrives when work starts and leaves when work finishes. A team member: a) manages time and resources in an effective manner b) respects and values the time of others c) understands and adapts to a general view of time (schedule events vs. relationships events) d) maintains a balance between work and social life **2. Collaboration**: impersonal relationships between team members: cooperation, trust, information exchange and positive thinking. A team member: a) demonstrates a balance between personal and professional roles and responsibilities b) works effectively to understand self and others c) provides consistent feedback to others regarding job performance d) works with others to enhance team commitment and collaboration	7-9

Table 4.4: Scale range for software project variables

Range Type	Description	Level
High	**3. Job Stability**: loyalty to the project and the organisation. A team member: a) creates a culture of collaboration and trust so that teams take risks and achieve performance goals b) enhances team loyalty and belonging c) creates trust among the different team members **4. Intercultural Intelligence**: ability to understand another culture. Questions sought to find out whether the team understands the feelings, values, and goals of others and is able to understand other cultures' world views. A team member: a) is able to understand other culture b) shows understanding of each other's culture c) recognises the abilities of team members d) is able to culturally express himself effectively **5. Team Experience**: Accumulated number of experiences. The questions sought to find out whether the team uses past experience to develop current or new projects. A team member: a) uses department's tools and equipment effectively b) uses skills and knowledge acquired from other projects c) is familiar with other team member's backgrounds d) worked on similar project previously **6. Communication**: Number of communication channels that have been created in the project; how information is distributed and sorted. The questions sought to find out whether the team communicates effectively. A team member: a) uses active listening and effective feedback process b) understands and uses effective relationship strategies to maintain trust within the team c) encourages the team to use their own leadership, judgement, and decision-making capabilities d) talks to everyone and uses teamwork to get things done **7. Reward Mechanism**: In this context means incentives mechanism: The questions sought to find out whether the team encourages and supports professional development and rewards. A team member: a) encourages and supports other team members in on-going professional growth opportunities b) recognises and rewards others for their achievements	7-9

Table 4.4a: Scale range for high scale team culture variables

Range Type	Description Definitions are similar to the high range questions "Some" or "Occasional" refers to approximately 50%	Level
Medium	**1. Timeliness**: A team member: a) occasionally reports to work or meetings late b) occasionally manages time and resources c) occasionally respects time of others' time d) sometimes maintains balance between work and social life **2. Collaboration**: A team member: a) demonstrates some balance between personal and professional roles and responsibilities b) works effectively to understand self and others sometimes c) provides some feedback to others regarding job performance d) sometimes works to enhance team commitment and collaboration **3. Job Stability**: A team member: a) creates some culture of collaboration and trust so that teams take some risks and achieve performance goals b) sometimes enhances team loyalty and belonging c) sometimes creates trust among the different team members **4. Intercultural Intelligence**: A team member: a) understands some of their shared culture b) has ability to understand some other cultures' world views c) recognises the abilities of some staff members d) is able to culturally express himself effectively sometimes **5. Team Experience**: A team member: a) sometimes uses department's tools and equipment effectively b) sometimes uses skills and knowledge acquired for the project c) is sometimes familiar with other teams' backgrounds d) sometimes works on similar projects **6. Communication**: A team member: a) uses active listening and effective feedback processes b) understands and uses effective relationship strategies to maintain trust within the team c) encourages the team to use their own leadership, judgement, and decision-making capabilities d) talks to everyone and uses teamwork to get things done **7. Reward Mechanism**: A team member: a) sometimes encourages and supports team in on-going professional growth opportunities b) sometimes recognises and rewards the team for their achievements	4-6

Table 4.4b: Scale range for medium scale team culture variables

<table>
<tr><th>Range Type</th><th>Description
Definitions are similar to the high range questions</th><th>Level</th></tr>
<tr><td>Low</td><td>1. Timeliness: A team member:
a) often reports to work or meetings late
b) manages time and resources badly
c) has no respect for the value of time of others
d) rarely maintains a balance between work and social life

2. Collaboration: A team member:
a) has no balance between personal and professional roles and responsibilities
b) has no effective understanding of self and others.
c) has no feedback to others regarding job performance.
d) has no team commitment and collaboration to enhance work

3. Job Stability: A team member:
a) does not create a culture of collaboration and trust so that teams take risks and achieve performance goals
b) does not enhance team loyalty and belonging
c) does not create trust among the different team members

4. Intercultural Intelligence: A team member:
a) shows no understanding of their shared culture
b) is unable to understand other cultures' world views
c) rarely recognises the abilities of team members
d) is unable to culturally express himself effectively

5. Team Experience: A team member:
a) does not utilise emerging technologies or technologies in proven business models
b) does not use departmental tools or equipment effectively
c) does not use skill and knowledge acquired from other projects
d) is not familiar with other team members' backgrounds
e) has not worked on similar projects

6. Communication: A team member:
a) uses passive listening instead of effective feedback processes
b) prevents empowerment or spot decisions
c) does not use effective relationship strategies to maintain trust within the team
d) does not encourage the team to use own leadership, judgement, and decision-making capabilities
e) does not talk to all team members and does not use teamwork to get things done

7. Reward Mechanism: A team member:
a) does not encourage and support team members in on-going professional growth opportunities
b) does not recognise and reward the team for their achievements</td><td>1-3</td></tr>
</table>

Table 4.4c: Scale range for low scale team culture variables

The measurement for leadership is similar to that of culture. Table 4.5 shows the three levels of measurements for leadership: high, nominal and low.

Range Type	Description	Level
High	1. **Interaction and Relationships:** creates learning experiences and treats team members with respect 2. **Decision-Making:** creates right decisions and consults with teams on direction. 3. **Ability to Motivate:** shares goals, provides appropriate instructions and gives support 4. **Understanding of Organisational Culture:** is able to understand and manage multicultural teams' backgrounds 5. **Active Thinking:** enhances team contributions and sets feasible targets 6. **Communication Skills:** assists and fosters good communication among team members, effectively and regularly	7-9
Medium	1. **Interaction and Relationships:** occasionally appreciates team members' good work 2. **Decision-Making:** occasionally makes good decisions and consults with team members to act on decisions and problems 3. **Ability to Motivate:** occasionally shares some of the goals, gives encouragement to achieve and provides some appropriate levels of direction and support 4. **Understanding of Organisational Culture:** is occasionally able to understand and manage multicultural team members' issues 5. **Active Thinking:** occasionally enhances team contributions and feasible targets 6. **Communication Skills:** sometimes assists communication among team members effectively	4-6
Low	1. **Interaction and Relationships**: does not appreciate any of the team members' work 2. **Decision-Making**: does not make many proper decisions or act quickly on decisions and problems 3. **Ability to Motivate**: does not share many goals and does not provide appropriate levels of direction and support 4. **Understanding of Organisational Culture**: does not have understanding and is unable to manage intercultural teams' backgrounds 5. **Active Thinking**: does not enhance team contributions and does not set feasible targets 6. **Communication Skills**: provides neither assistance to nor communicates with team members and rarely has feedback on team issues	1-3

Table 4.5: Scale range for leadership variables

4.9 Summary

From the survey of state-of-the-art software project cost measurement, some major issues causing misunderstanding and misapplication of project measurement were identified. Various cost measurement approaches were investigated in order to integrate the different aspects of measurements. To provide a common protocol, a framework was developed to capture explicitly the relationships among apparently diverse project measurement activities. A distinguishing feature of the framework is its use as the basis for exploring a means of measuring software project cost within the UAE context by incorporating leadership and cultural factors. This framework is supported by an ontology that was constructed to unify the semantics of project parameters. The ontology supports a consistent measurement strategy and establishes a common measurement protocol. The SEEOS was developed for use in estimating software project cost. The automated SEEOS assists in the estimation of software project effort and cost. The final model is an augmented CBR system for effort estimation.

CHAPTER 5

The Cost Estimation Model and its implementation

5.1 Introduction

This chapter introduces the software effort estimation ontology system (SEEOS) to assist in the estimation of a software project's effort and cost. The system incorporates specific cultural characteristics, factors, and issues that are identified as influencing cost estimation.

The following gives a summary of the different sections of this chapter. Section 5.2 discusses the analogy method, and Section 5.3 presents the steps of estimation by analogy. Section 5.4 presents the augmented CBR approach. Section 5.5 presents the model variables, and Section 5.6 presents the approach used in the cost estimation model. Section 5.7 discusses the ontology-based system, and Section 5.8 discusses model validation. Section 5.9 describes system implementation, and Section 5.10 discusses the system architecture and layers. Sections 5.11 and 5.12 discuss validation and the user interface and a summary is presented in section 5.13.

5.2 The Analogy Method

Prediction methods used in estimation techniques fall into the following classes:

- expert judgement (or experience–based estimation)
- statistical models such as regression
- Case Based Reasoning (CBR)

Expert judgement may be rather ad hoc. It is used when no historical data are available. The application of regression methods requires rather strict normality assumptions on the data and very often advanced manipulation and transformation of the variables. When it comes to cases where the response variable (the dependent variable) is a ratio that is not normally distributed and the factors (independent variables) are in a nominal or in an ordinal scale, the equations resulting from its application are very difficult to interpret and use for prediction. Familiarity is needed with the advanced demands of the method. On the other hand, CBR, which includes estimation by analogy, can be easily applied to any type of variables and can be interpreted in a straightforward manner.

5.3 The Steps of *Estimation by Analogy (* EbA)

Estimation by Analogy (EbA) is a very simple, but powerful method. It has been used in many different applications and particularly in classification tasks. The key idea behind the EbA is that similar input data vectors have similar output values. A certain number of nearest neighbours are sought according to a distance metric and their corresponding output values are used to determine the output approximation. The estimation of the outputs is calculated by using the average of the outputs of the neighbours (analogies). The EbA method is described briefly by Stamelos and Angelis (2001). It is a procedure consisting of three steps. First, the new class for which the project effort is to be estimated is characterised by a set of attributes common to the ones characterising previous projects in a historical database. Second, one or more similar projects (neighbours or analogies) from the dataset are found according to a predefined distance metric. Finally, the values of the neighbour projects are used to produce the estimate (usually by computing their mean).

A proposed solution is based on historical similar cases (projects) extracted from a database containing, in addition to the actual effort, information on quantitative and qualitative attributes from previous projects. The new case (project) is characterised according to the attributes (factors) used and then placed into an historical dataset. The data structure is shown in Figure 5.1:

	Effort	*Attribute 1*	*Attribute 2*	...	*Attribute k*
Case 1	E_1	X_{11}	X_{12}	...	X_{1k}
Case 2	E_2	X_{21}	X_{22}	...	X_{2k}
....	...	...	...	...	...
Case n	E_n	X_{n1}	X_{n2}	...	X_{nk}
New Case	Unknown	Y_1	Y_2	...	Y_k

Figure 5.1: The data structure (Angelis and Stamelos, 2000)

Since the datasets usually contain mixed-typed variables (nominal, ordinal and continuous), a distance metric, as suggested by Shepperd and Schofield (1997), is used. In the following, it is assumed that the new project is represented by a vector of attributes $(Y_1,...,Y_k)$ and every project i in the database by the vector $(X_{i1},...,X_{ik})$. The distance is then computed (see Figure 5.2).

$$Sim(C_1,C_2,F) = \frac{1}{\sqrt{\sum_{j \in P} Feature\_Dissimilarity(C_{1j},C_{2j})}}$$

$$F = Set\ of\ features$$

$$C_1,C_2 = Cases$$

$$\text{Feature\_Dissimilarity}\,(C_{1j},C_{2j}) \begin{cases} (C_{1j} - C_{2j})^2 & \text{Features are numeric} \\ 0 & \text{Features are categorical and } C_{1j} = C_{2j} \\ 1 & \text{Features are categorical and } C_{1j} \neq C_{2j} \end{cases}$$

Figure 5.2: Nearest neighbour algorithm (Shepperd and Schofield, 1997)

A statistical analysis of the data is needed first. Usually, regression models give good results, but the variables have to be transformed at the outset. For example, the effort should be logarithmically transformed. Dissimilarity measures (Shepperd and Schofield, 1997; Angelis and Stamelos, 2000) can be computed when there are mixed (numerical and categorical) data. Another advantage of the measure is that it takes into account the missing values of the projects as well; this is appropriate for missing data. In such a case, the distances are usually calculated using only the available attributes. A more advanced method is to estimate the missing value, although this is difficult and often very risky. In principle, missing data can be estimated in such a way that the overall output will not change (no increase or reduction) by adding such data (thus, dummy data). Figure 5.3 shows the algorithms used to measure similarity and dissimilarity. The mixed data (numerical and categorical) were calculated using the unweighted Euclidean distance employed by previous researchers (Shepperd, 2003; Angelis and Stamelos, 2000). This approach was used earlier with encouraging results in cost estimation and other areas (Shepperd and Schofield, 1997).

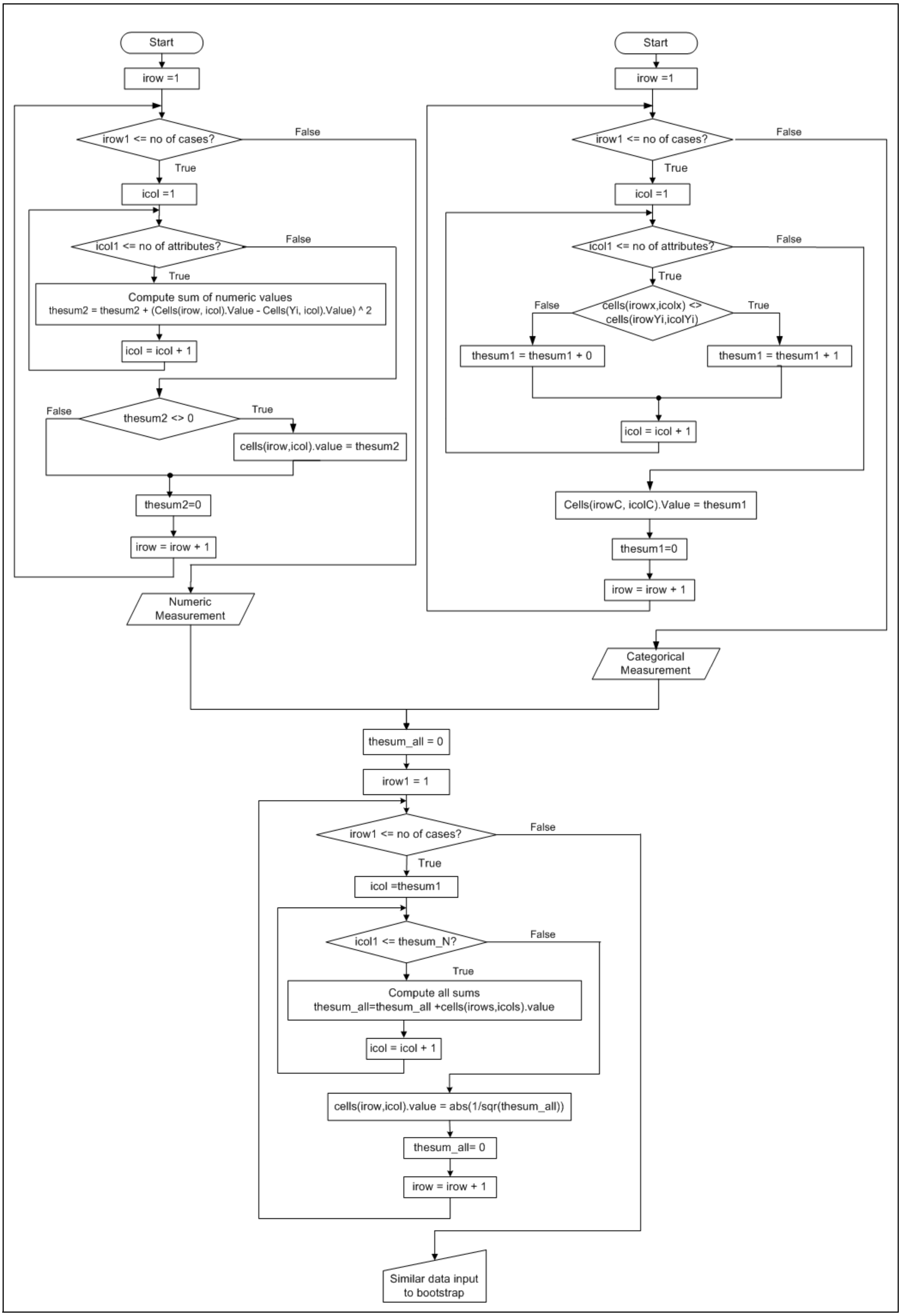


Figure 5.3: Distance and the similarity measures algorithm

An important decision for EbA is the number of analogies (usually a small number) that has to be combined in order to evaluate the dependent variable. After selecting the distance metric and the number of analogies, it needs to utilise a statistic that will combine the actual effort (dependent variable) of the analogies. In this study, the mean was used for such a combination while the number of analogies was decided after an extensive calibration procedure. This is discussed in more detail in Chapter 6.

According to the theoretical and the experimental findings (Atkinson and Shepperd, 1994), a small number of projects should be compared (1-3 projects). However, the case may be different when studying software effort estimation in governmental departments since the above-mentioned parameters are similar across many governmental departments which share the same culture and economical strength. When the parameters' variance is small among governmental departments, more projects should be taken into consideration leading to greater accuracy in estimation. However, in cases where the variance is high, then a small number should be selected, perhaps 2-3 (Stamelos and Angelis, 2001). A simple averaging arithmetic mean is the most appropriate method in calculating the project estimate since the projects are similar. However, in cases where the difference is high, then the median method may be used.

5.4 Case-Based Reasoning (CBR)

The CBR technique is an approach to solve new problems by recalling a previous similar situation and by reusing information and knowledge of that situation (Aamodt and Plaza, 1994). The solution of a case with the most similar match is adapted to solve the new problem (Kolodner, 1993).

The CBR procedure is similar to an ontology-based representation where the ontology defines the semantics of the project parameters and their relationships and establishes shared concepts between different projects. Hence, common features of CBR maintain past historical experiences in memory to find similar cases for the new problem, also adapting existing solutions to match current situations and refine the memory for future new cases. The difficulty for CBR and other approaches is in retrieving more useful cases where irrelevant and misleading features are used in a case but relevant features are ignored (Kolodner, 1992; Shepperd, 2003). These features represent actual cases so it is more important to keep such features that represent the case because knowing which features are useful is not always obvious. Therefore, an ontology can determine a set of measurement protocols and definitions of the attributes and factors that are more representative of a case.

The augmented CBR process consists of the following stages (see Figure 5.4) to estimate software cost:

- given n projects (cases) characterised by p features
- a new project characterised by p features
- retrieve most similar projects in the p-dimensional space to the new project
- define culture and leadership values for the new project
- assign measurements to the new case using the ontology-based system
- add the new case to the case base

In particular, completed projects are characterised in terms of a set of p-features. The new active project is also characterised in terms of the common p-features for which an estimate is required. The features (attributes) usually represent measures which influence effort behaviour such as size measures and cultural factors. The similarity between the new case and the other cases in the p-dimensional feature space is measured, and the most similar cases (or projects) are used, possibly with adaptations to obtain a prediction for the new case. This process can be achieved by using the nearest neighbour techniques (Shepperd, et al, 1996; Shepperd and Schofield, 1997).

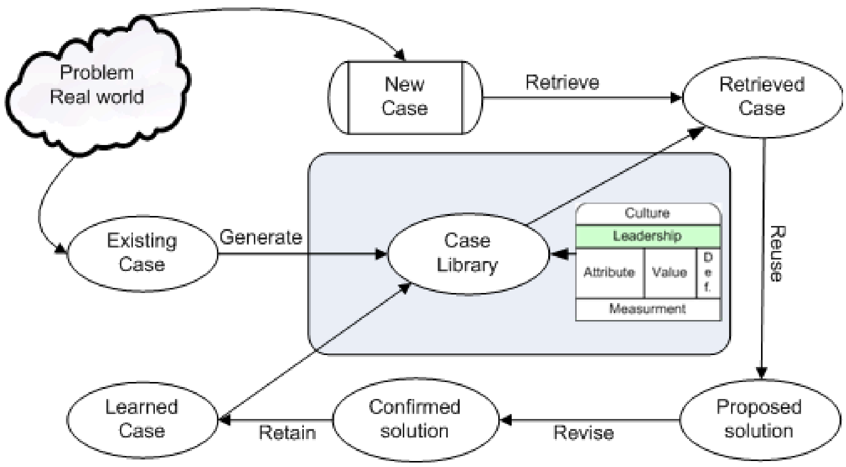


Figure 5.4: A general framework of the CBR process

The major difference between the SEEOS analogy and CBR analogy is that CBR does not take into account culture and leadership. The inclusion of culture and leadership would be a definite enhancement. The model agrees with CBR in cases without culture and leadership. Thus, adding these new attributes removes some of the limitations of CBR (see Figure 5.5).

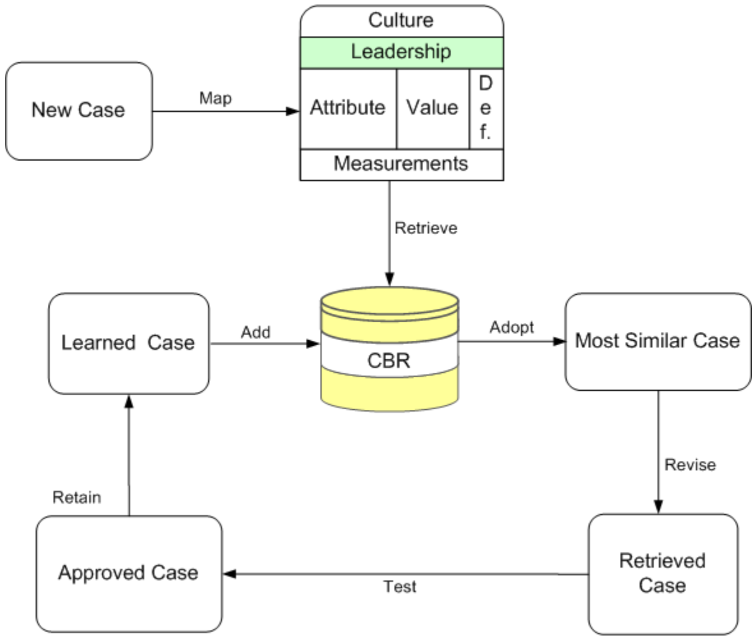


Figure 5.5: The augmented CBR

CBR allows the construction of a library of cases, or experiences, which are described according to a fixed set of descriptors called the domain model. The domain model has parameters (such as "project size" or "line of business") and values (such as "budget", "government services", etc.). Afterwards, relevant cases are recalled by specifying a set of Parameters-values. For example (see Table 5.1), a simple query result could be:

Project Size for HR / Payroll	Function Points (value)
Reports	800
Screens	240
Tables	321

Table 5.1: Human Resources /Payroll CBR query

Case searching is used to compare and match cases from the database in order to find similar cases. The search is based on key attributes and features. Case indexing techniques are used, especially, for large databases (Patterson, et al., 2005). Thus, CBR provides a way of archiving experiences and quickly retrieving appropriate experiences when needed. As the importance of certain values can be emphasised over others, the search queries can be precise. For example, the project case base (see Table 5.2 and Appendix E) which contains the significant data items from main projects, includes organisations, system management, project type, duration, estimate cost, actual cost, application type and actual effort. This is done to make the similarity search easy to conduct, especially when the data and contexts vary. CBR can handle cost estimation problems efficiently. The fact that CBR is derived from human reasoning has given this approach a significant advantage in software development cost estimation. CBR is simple and flexible compared to numerical models. Another advantage of CBR is that no expert is required because everything is based on the analogy gleaned from prior cases and applied to new ones. Also, CBR can handle quantitative and qualitative data (Snell, 1997; Angelis and Stamelos, 2000).

Field Name	***Data Type***	***Field Size***	***Remark***
OrgNo	Text	4	*Key*
Project_ID	Text	4	*Key*
SYS_MGT_ID	Text	3	*Key*
DEV_ID	Text	3	*Key*
Project_Type	Text	100	*Data*
Start_Date	Date/Time	dd-mm-yy	*Data*
Finish_Date	Date/Time	dd-mm-yy	*Data*
Estimation Cost	Currency	--	*Data*
Actual Cost	Currency	--	*Data*
Application_ID	Text	3	*Foreign Key*
Leadership_ID	Text	3	*Foreign Key*
Actual Effort	Number	Long Integer	*Data*

Table 5.2: Metadata that describe a project

5.5 Presentation of the Model Variables

In an effort to derive an improved cost estimation model that is sensitive to local customs, a survey was used to identify which parameters impact the cost estimation model development. These parameters were not considered in previous studies although they impact on the cost estimation model development. The identified parameters have been categorised into seven groups (see Figure 5.6): organisation's lines of business, application type, organisation type, organisation culture, project leadership, project technical environment and year of project completion. Determining the appropriate parameters for defining the applicable cost estimation model is one of many major tasks. The new parameters and the approach for developing the cost estimation model will be presented subsequently.

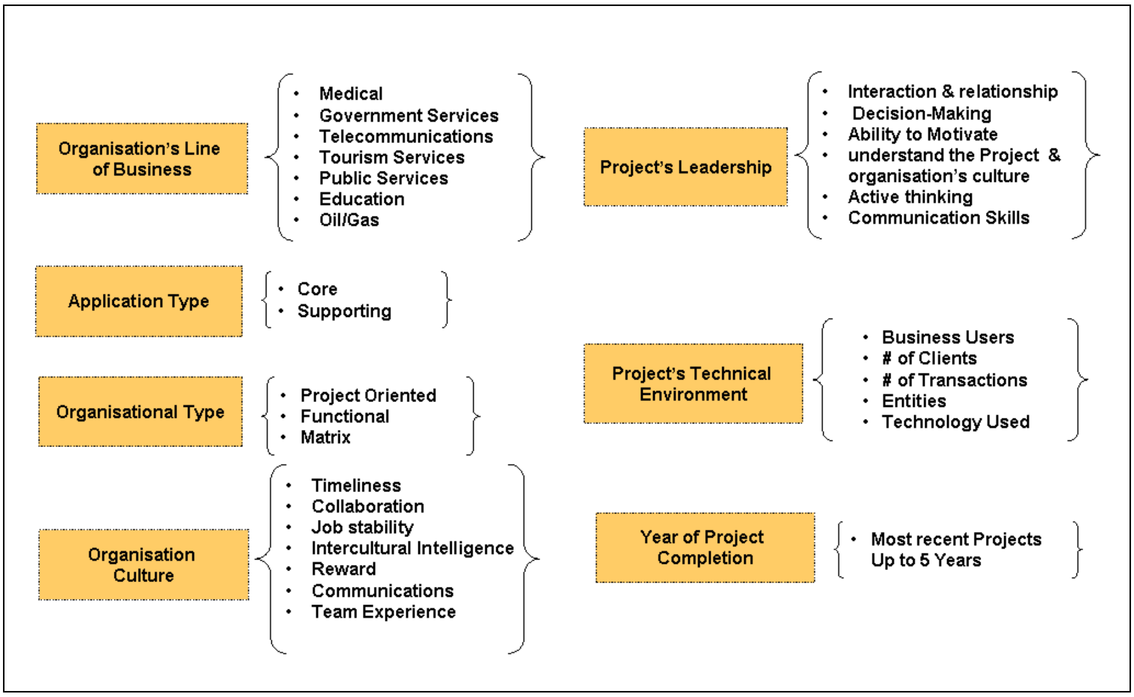


Figure 5.6: Model parameters used

While investigating the parameters that impact the software effort cost estimation in governmental departments, the case-based modelling process was derived (see Figure 5.7). This process identifies the stages and the parameters that impact cost estimation and should be taken into consideration when performing software effort estimation.

Figure 5.7: Proposed effort estimation process

5.6 Cost Estimation Module Approach

The dataset was split according to application type – "supporting" and "core" applications. Supporting applications are the systems which support the internal (shared) services in any organisation. These applications are not linked directly to the organisation mission and vision, rather they enhance the efficiency, effectiveness and the performance of the supporting services. Those systems share similar features across the government departments.

Core applications exist to help to achieve the mission and vision of the organisations and to satisfy their core purpose. The features of these applications are unique. Organisations with a similar line of business could share similar features.

Determining the project manager's level of authority is important. Projects with different levels of project authority cannot be considered similar even if the other parameters are similar. Only projects that are developed in a similar organisational structure may be taken further by determining similarity with the other parameters.

Determining similar projects that operate in a similar organisational culture and determining an acceptable level of similarity among projects with different organisational cultures is challenging. For example, project A and project B have similarity in decision-making, ability to motivate team members, communications and project managers' experience; they are different in organisational culture, even though they are similar with respect to other parameters. The difference in understanding culture can be substituted by a buffer time. The magnitude of impact needs calibration after further studies. For example, at 4-6 on the scale of variables, 10% more man-days need to be added to the new project to accommodate individual differences (Kitchenham et al., 2002). The weight on the scale is assigned according to the effectiveness of each parameter in the process.

As a result, the culture and leadership attributes were given values (1-9 or low - high) from an accumulated set of decisions about each attribute and the average of sub-unit scores for each attribute was found. The common leadership and cultural factors were selected, studied and measured from several perspectives (see Figure 5.8).

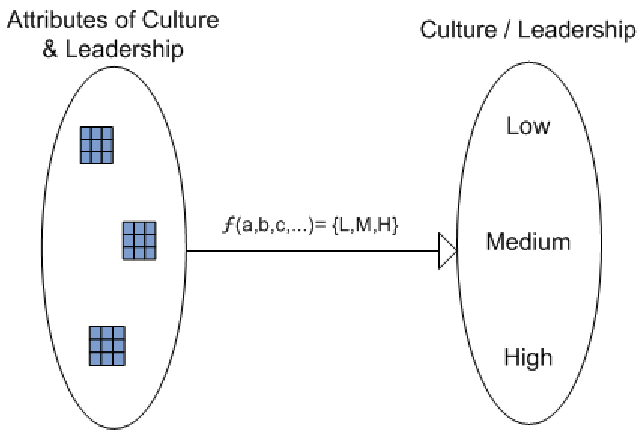


Figure 5.8: Leadership and culture scale values

Various generic attributes such as personality traits, power relationship and behaviour changes were observed. Also a parallel approach to attributes that influence the estimation process such as technology used and project complexity were also examined (see Figure 5.9, 5.10). All attributes may have a postitive or negative impact on the software development and cost estimation.

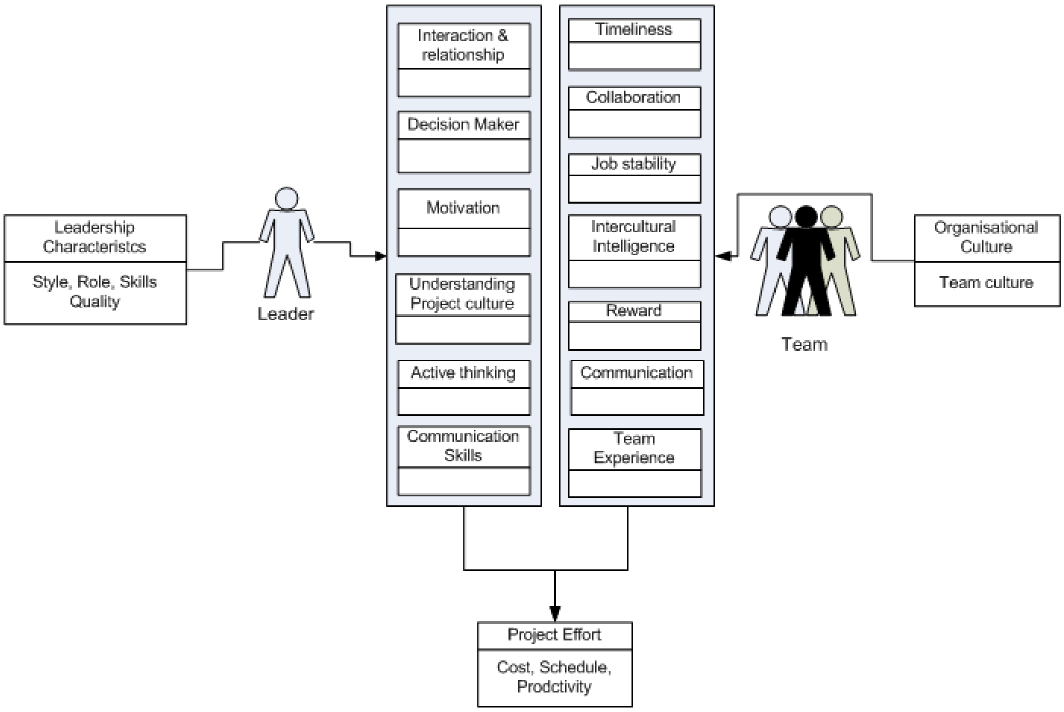
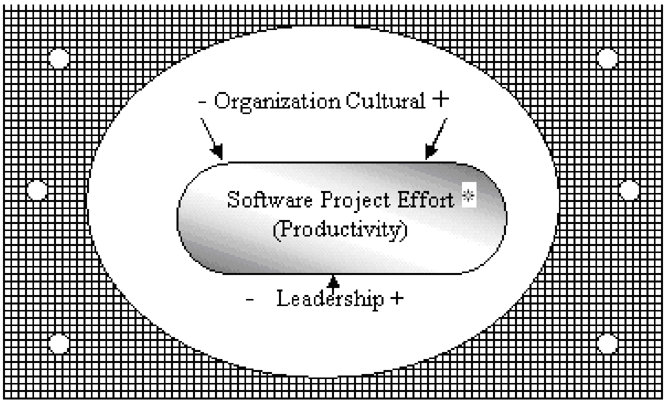


Figure 5.9: Conceptual model of culture and leadership



Key: * important + pro - con ○ other attributes (technology used and project complexity)

Figure 5.10: Attributes that affect cost estimation

Each of the items describes a certain leadership behaviour, characteristic or effect that a leader may have on the organisation. Here, the leaders in the organisations were at two levels: the project team could evaluate their executive manager; project manager could evaluate team members. The variables were included in the model for the similarity among software projects. A a query system which captures this case was developed (see Figure 5.11).

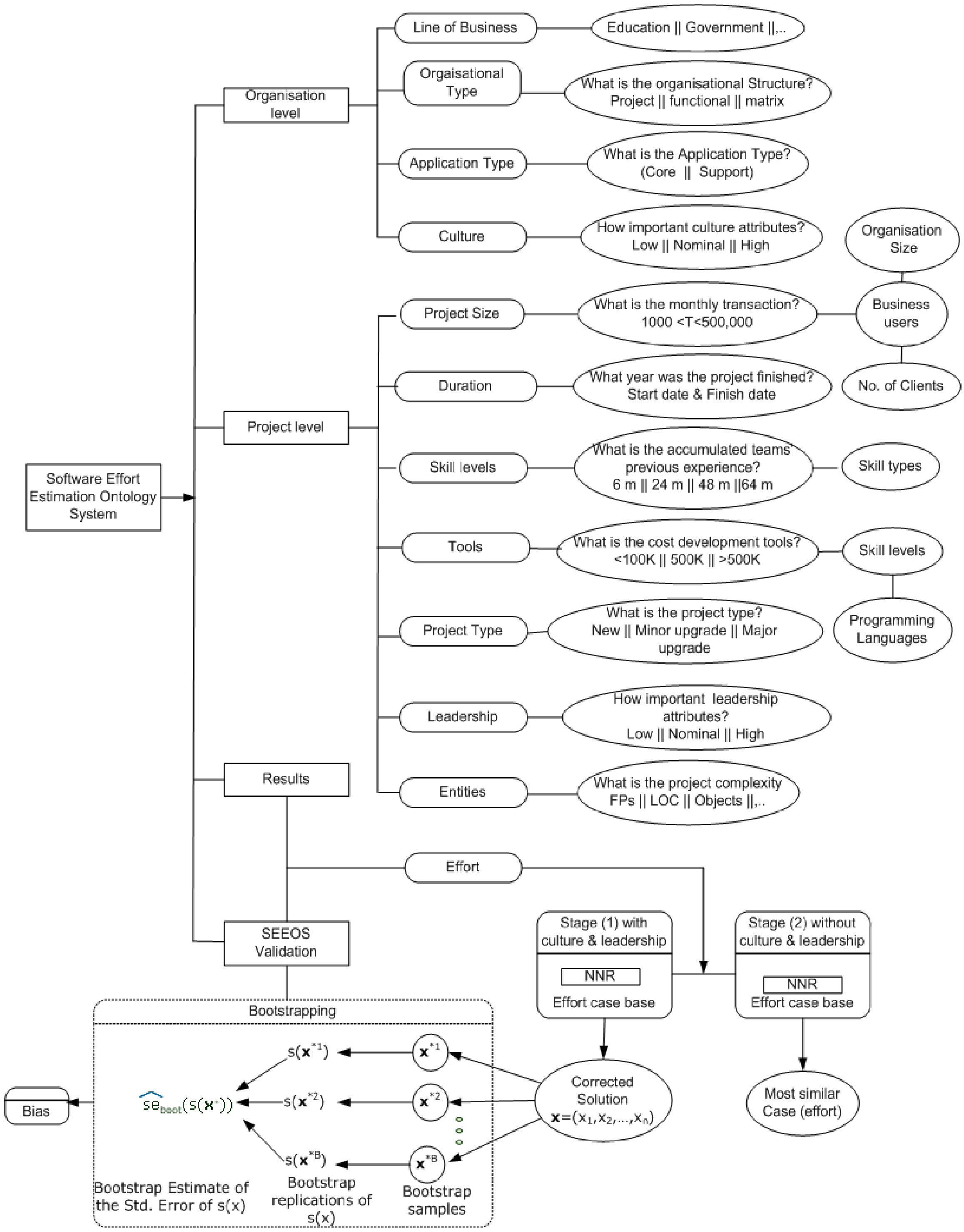


Figure 5.11: The structure for the query system

The model was tested on a number of governmental development projects in order to determine its accuracy and appropriateness. This work explored various ways of classifying the existing software cost and effort estimation techniques, providing an overview of each category and discussing in more depth with a comparison of the most popular cost models. The results of the work were then analysed and compared with existing CBR tools, e.g. ANGEL (Shepperd and Schofield, 1997).

As the system was designed to provide an environment for examining the feasibility and validity of the proposed model, the developed version required further improvement. In order to offer a system to search for all possible feature subsets, there was a need to be able to cluster cases by business domain and complexity. Because these particular projects were from various domains and shared few projects, clustering the projects presented some serious difficulty. The complexity of the projects was identified by including it with the line of business that made some attributes depend on other attributes. For example, it was difficult to measure the similarity of core systems in a different domain. Another limitation in the research focused on the analogy and ontology to retrieve and reuse processes of the CBR cycle. This is, as yet, has not been implemented in the current system and will be the subject of further study.

This system was implemented using Java and a relational database management system (see Figure 5.12). The data were collected within UAE organisations using an online system that has been developed as part of this project. The relational database was used to collect new projects in one central repository and the entire application is a stand-alone system.

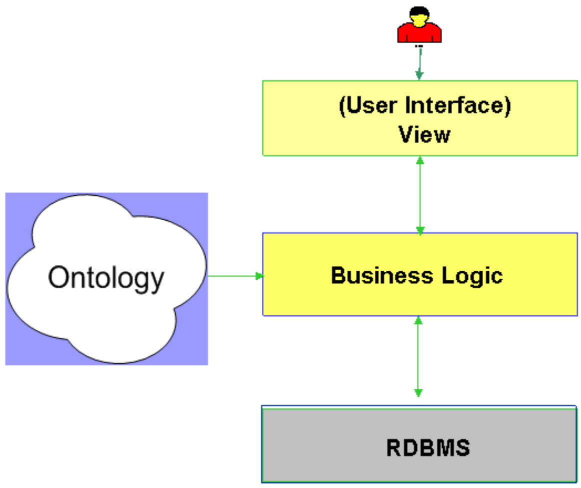


Figure 5.12: Three -Tier system architecture

5.7 The Ontology-Based System

The ontology enables project managers to elicit more accurate software project features that are semantically suitable for these requirements. It provides a common understanding of software parameters and their semantics. A formal description of the concepts and relationships is provided for each attribute and definition. This ontology was discussed previously in Chapter 4.

5.8 Model Validation

The use of bootstrap in estimating the effort involved in a given project is quite challenging due to difficulties in describing quantitatively variables such as effort, which are rather abstract. The proposed method requires a minimal mathematical background and very few assumptions. Instead, it relies heavily on computing power. The approach is general and yet simple and intuitive. It is a statistical technique introduced by Efron in 1979 (Efron, 1992; Efron and Tibshirani, 1993), aiming at measuring statistical accuracy such as bias and prediction error. Also it handles complex data structures such time series and regression models requiring little theoretical knowledge.

The technique of bootstrapping is implemented in the following 3-stage procedure (Shepperd and Schofield, 1997; Stamelos and Angelis, 2001).

1. Extraction of similar cases: extract (from a database where the information is stored) the historical cases that are most similar to the current one according to a selected metric.
2. Estimate the response variable “effort” for the new case (by EbA) based on the cases selected.
3. Estimate the precision and carry out validation.

a) Bootstrap estimate of the standard deviation (see Figure 5.13a):
   - Store the data (effort) of the selected cases in a convenient file.
   - Select (with replacement) the bootstrap samples.
   - Determine the bootstrap replicates of the median (or the mean).
   - Compute the standard deviation of the bootstrap replicates of the median (or the mean).

b) Bootstrap estimate of the bias and carry out validation (see Figure 5.13b).
   - Compute the bootstrap replicates of the bias for the median (or mean) as the difference between each bootstrap replicate of the median (or mean) and the sample median (or mean) of the dataset used.
   - Compute the bootstrap estimate of the bias as the average of the bootstrap replicates of the bias. The mean bias either shows the over estimation (+) or underestimation of the effort (-). A positive value of the mean bias represents overestimation and a negative value represents underestimation.
   - Validate by correcting for the bias according to its (+/-) sign.

Regarding the evaluation of the predictive accuracy for the EbA method, the jack-knife procedures were adapted from Kohavi (1995) and will be discussed in Chapter 7.

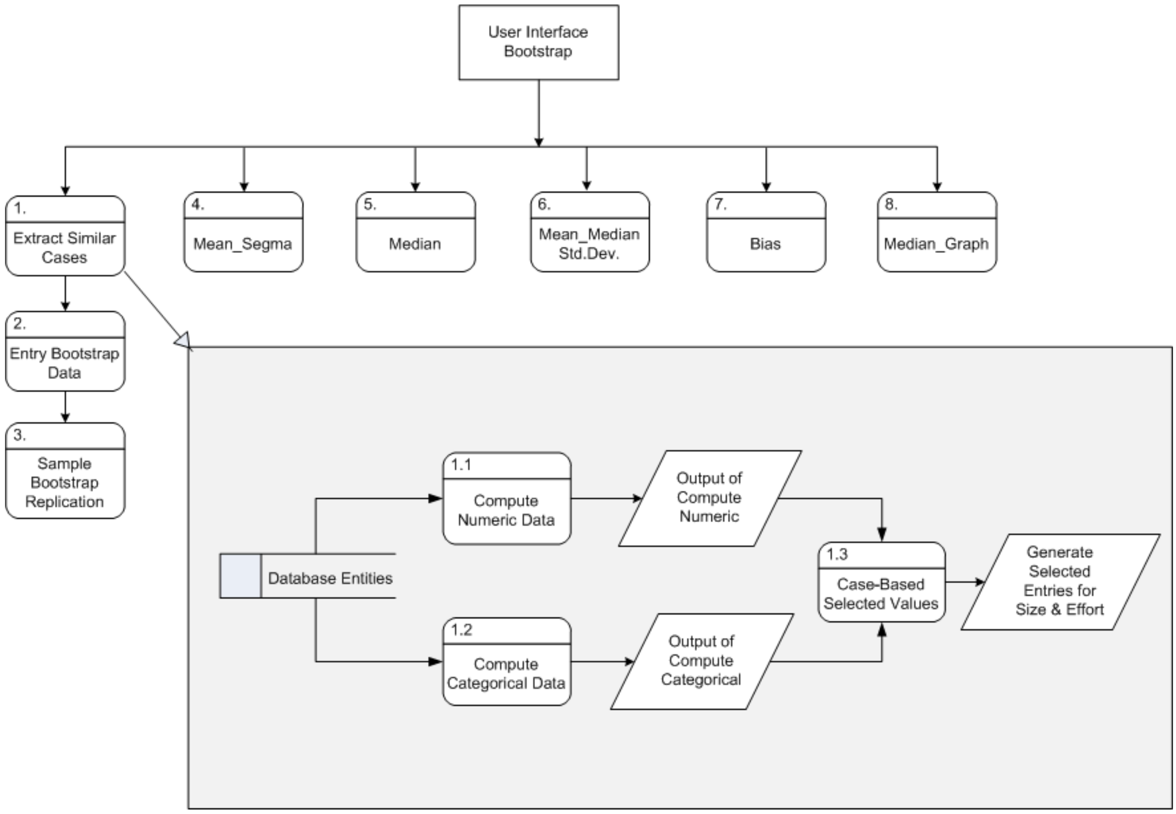


Figure 5.13a: Main context diagram of bootstrap process

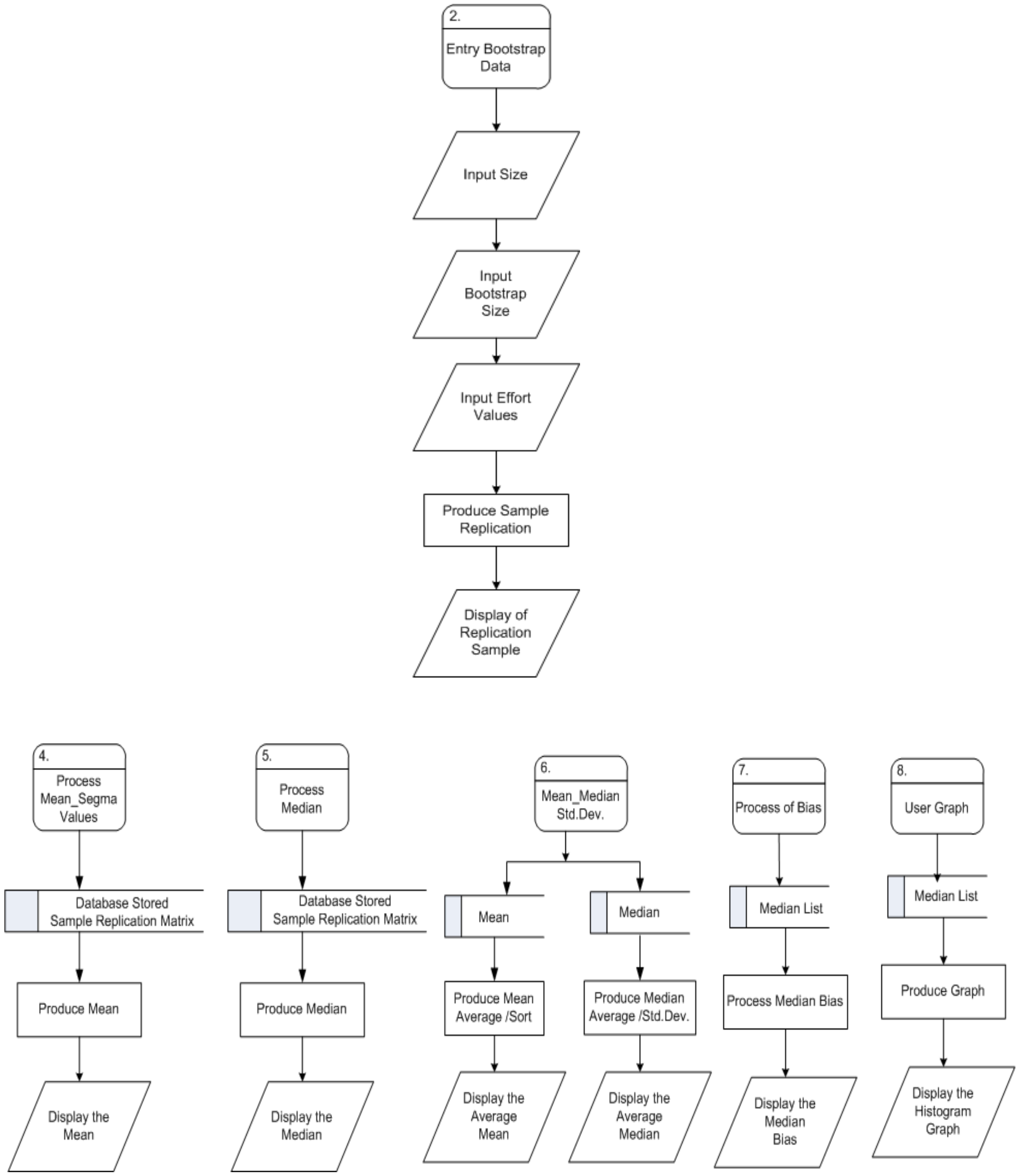


Figure 5.13b: DFD (Data Flow Diagram) for each activity of Bootstrap process

5.9 Implementation of the System

A software tool, called SEEOS (software effort estimation ontology system) that supports the application of an analogy-based method was implemented. This tool provides a flexible interface that allows users to experiment with different project characteristic options. The main functions of SEEOS are the following: (1) defining comprehensive attributes for a project; (2) defining attributes, characteristics and measurement protocol; (3) providing the choice of options to be considered such as cultural factors and leadership; (4) determining which attributes are available to provide better accuracy; and (5) generating most similar projects for the required estimate.

SEEOS consists of three subsystems; the analogy subsystem to find the most similar projects, the online subsystem used by different organisations to input projects data, and the bootstrap subsystem to validate the projects result. The system has been implemented using Java, JSP, Serlvet and a relational database management system.

5.10 System Architecture

A three-tier architecture is used in this project to structure the implementation. The three-tier architecture allows different tiers to be developed in different languages, such as a graphical user interface language or light internet clients (HTML, applets) for the top tier; Java, C, or C++ for the middle tier; and SQL for much of the database tier (Edelstein, 1994). The three-tier architecture of the SEEOS consists of a presentation layer, a business logic layer, and a database layer. Figure 5.14 illustrates the SEEOS three-tier architecture.

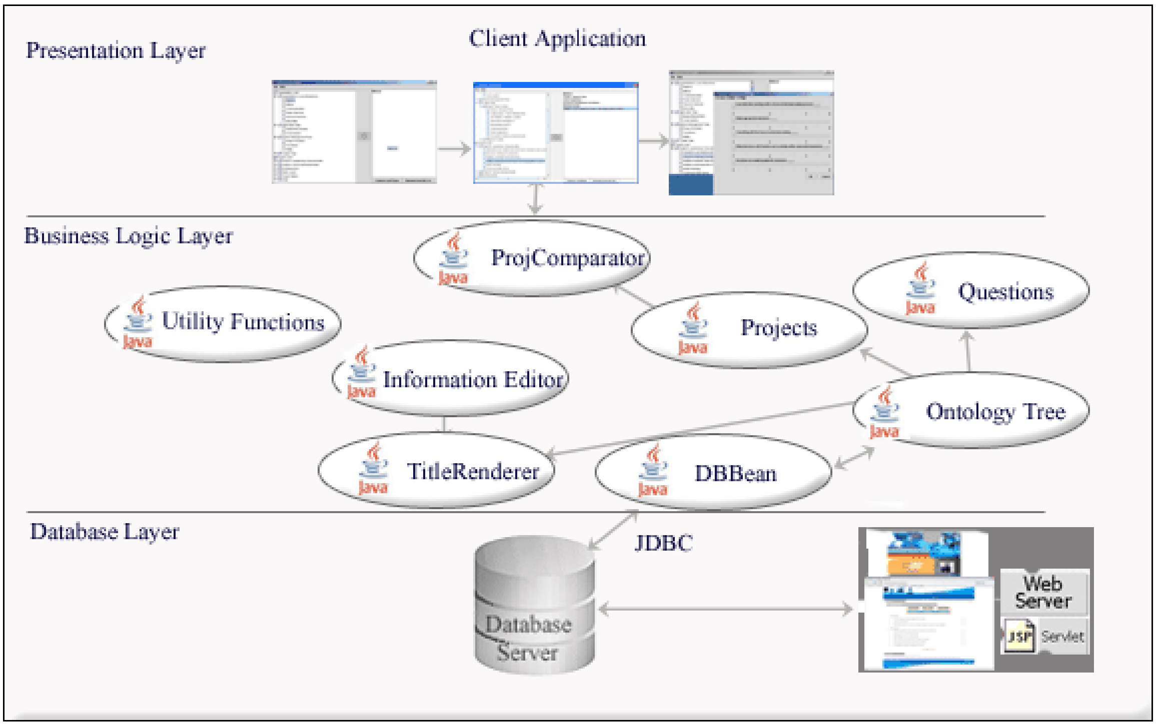


Figure 5.14: SEEOS system architecture

5.10.1 SEEOS Presentation Layer

The main SEEOS presentation layer consists of two panels; the right side panel and the left side panel (see Figure 5.15). The left side panel shows the project's entities along with their attributes, descriptions and values. It consists of a tree structure of different selection keys related to the organisation and projects. On the other hand, the right panel shows the selected entities to be estimated by the project manager. It consists of a list which will be populated by the selection keys on the left panel by pressing the "Right Arrow" button located between the two panels. In addition, the right panel has two buttons which are "Compare and Show" and "Remove from the List".

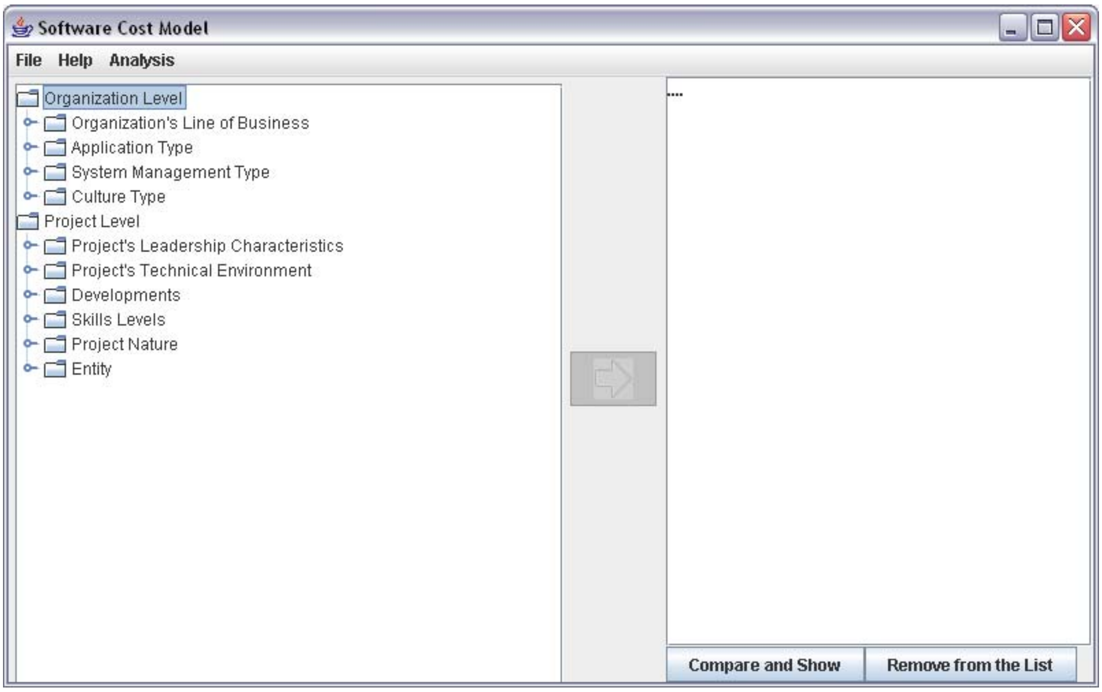


Figure 5.15: SEEOS user interface

To find the most similar projects, the project manager can select the project's attributes from a particular leader in the tree structure; he then clicks the "Right Arrow" button to transfer the project attributes to the right panel. The project manager has the option to remove attributes from the list by clicking the "Remove from the List" button in the right panel. Figure 5.16 shows an example of keys selected from a list.

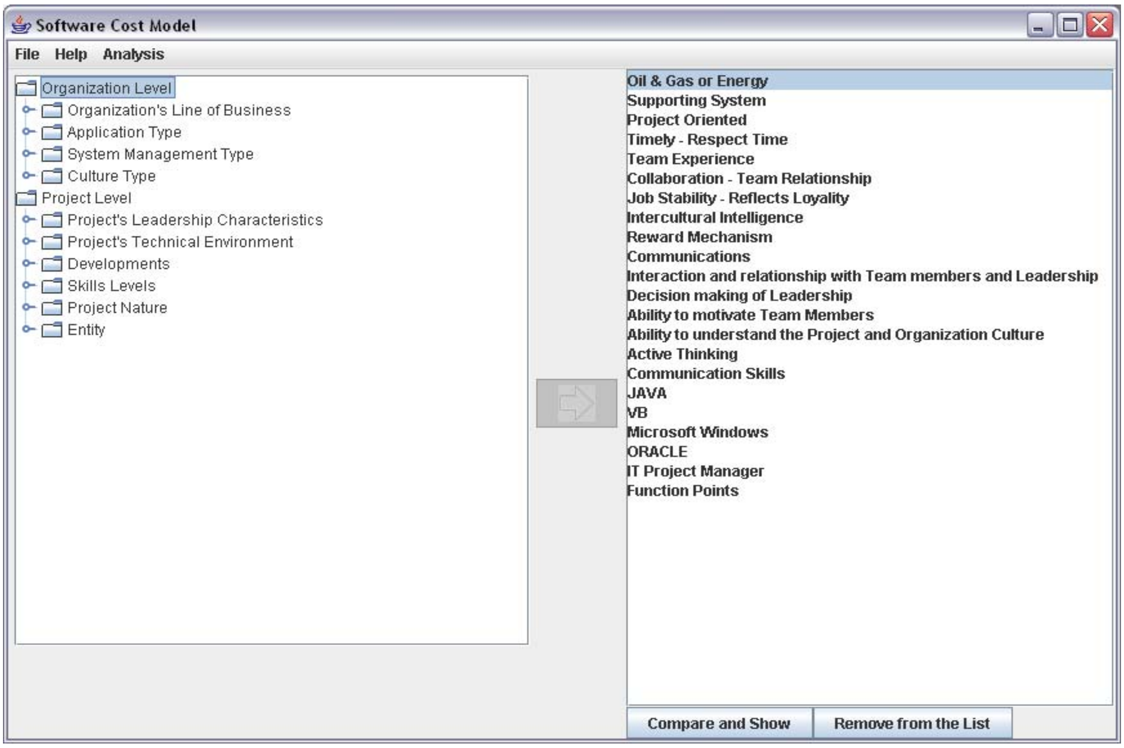


Figure 5.16: SEEOS keys selected

There are uncertainties in the way various project terms, variables and factors are interpreted. Two projects that may seem similar may indeed be different in a critical way. Moreover, the uncertainty in assessing similarities and differences means that two different analysts could develop significantly different views and effort estimates.

The project manager is able to select similar existing software projects based on well-understood project similarity features. The SEEOS system establishes a set of common project parameters between different projects and provides a common understanding (ontology) of project parameters and their semantics. It accomplishes this by allowing the project manager to give more semantic content to the new project attributes by selecting the attribute’s value and measurement. Figure 5.17 shows an example of keys surveyed to be filled.

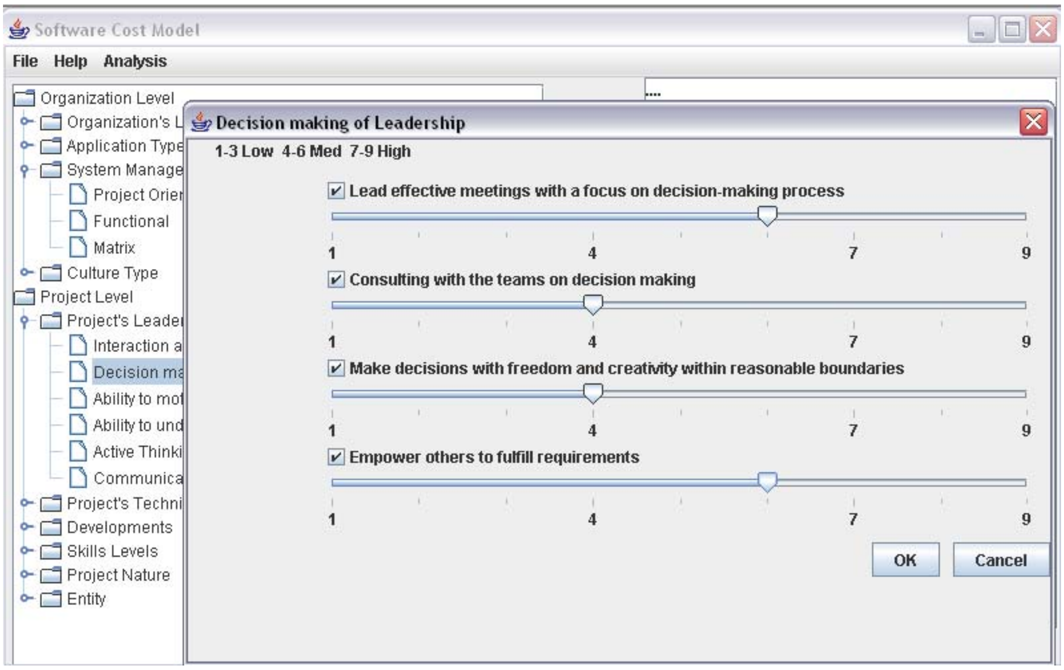
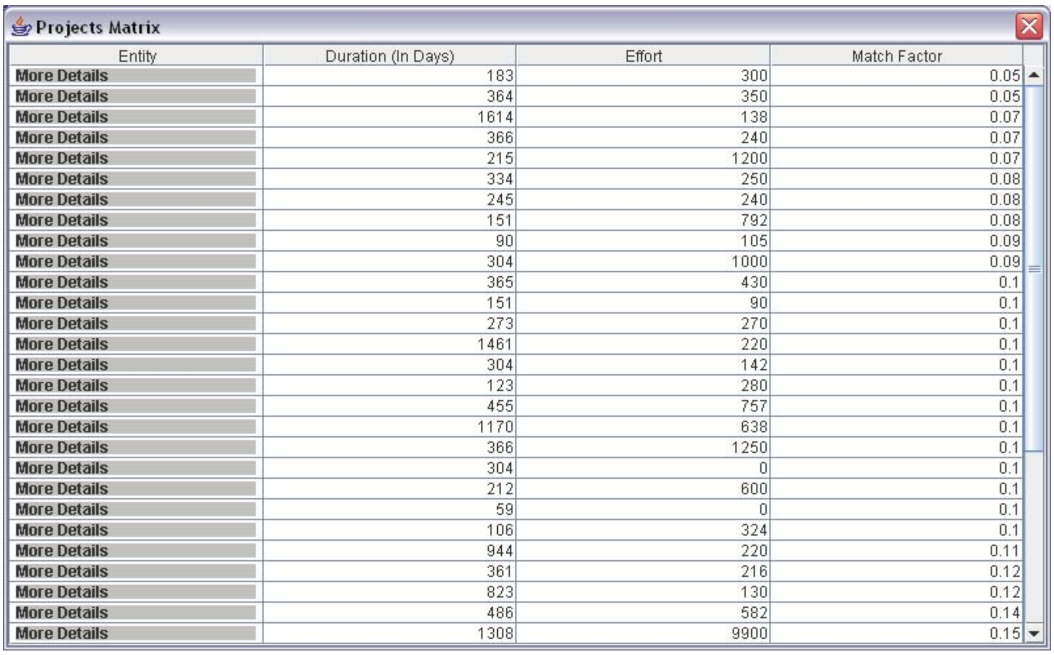


Figure 5.17: SEEOS Questionnaires

After populating the list with different keys related to a project, the project manager can compare these keys with the ones in the database for different projects by clicking “Compare and Show”. A dialog box will appear showing the comparison of results between the new project and projects in the database. Figure 5.18 shows an example of results.


Projects Matrix

Entity	Duration (In Days)	Effort	Match Factor
More Details	183	300	0.05
More Details	364	350	0.05
More Details	1614	138	0.07
More Details	366	240	0.07
More Details	215	1200	0.07
More Details	334	250	0.08
More Details	245	240	0.08
More Details	151	792	0.08
More Details	90	105	0.09
More Details	304	1000	0.09
More Details	365	430	0.1
More Details	151	90	0.1
More Details	273	270	0.1
More Details	1461	220	0.1
More Details	304	142	0.1
More Details	123	280	0.1
More Details	455	757	0.1
More Details	1170	638	0.1
More Details	366	1250	0.1
More Details	304	0	0.1
More Details	212	600	0.1
More Details	59	0	0.1
More Details	106	324	0.1
More Details	944	220	0.11
More Details	361	216	0.12
More Details	823	130	0.12
More Details	486	582	0.14
More Details	1308	9900	0.15

Figure 5.18: SEEOS Results

The “Duration” column shows the number of days taken to complete that particular project. The “Effort” column shows the number of man-days involved in completing that project; and the last column, the “Match Factor”, shows the calculated similarity value of that project with that created by the user.

5.10.2 SEEOS Business Logic Layer

The SEEOS business logic layer consists of seven Java classes (see Figure 5.19). The “OntologyTree” is the main class for drawing the main window with a tree of different elements and a list box with different command buttons. “DBBean” is a class used for the database management. “InformationEditor” is a class used to add buttons in particular cells. “ProjComparator” is a class used to sort the values of vectors in ascending order. “Projects” is a class used to store all the values against each project in the Database. “Questions” is a class used to store question and answer values. “TitleRenderer” is a class which keeps track of cells of tables. “UtilityFunctions” is also a class consisting of miscellaneous methods providing different functionalities.

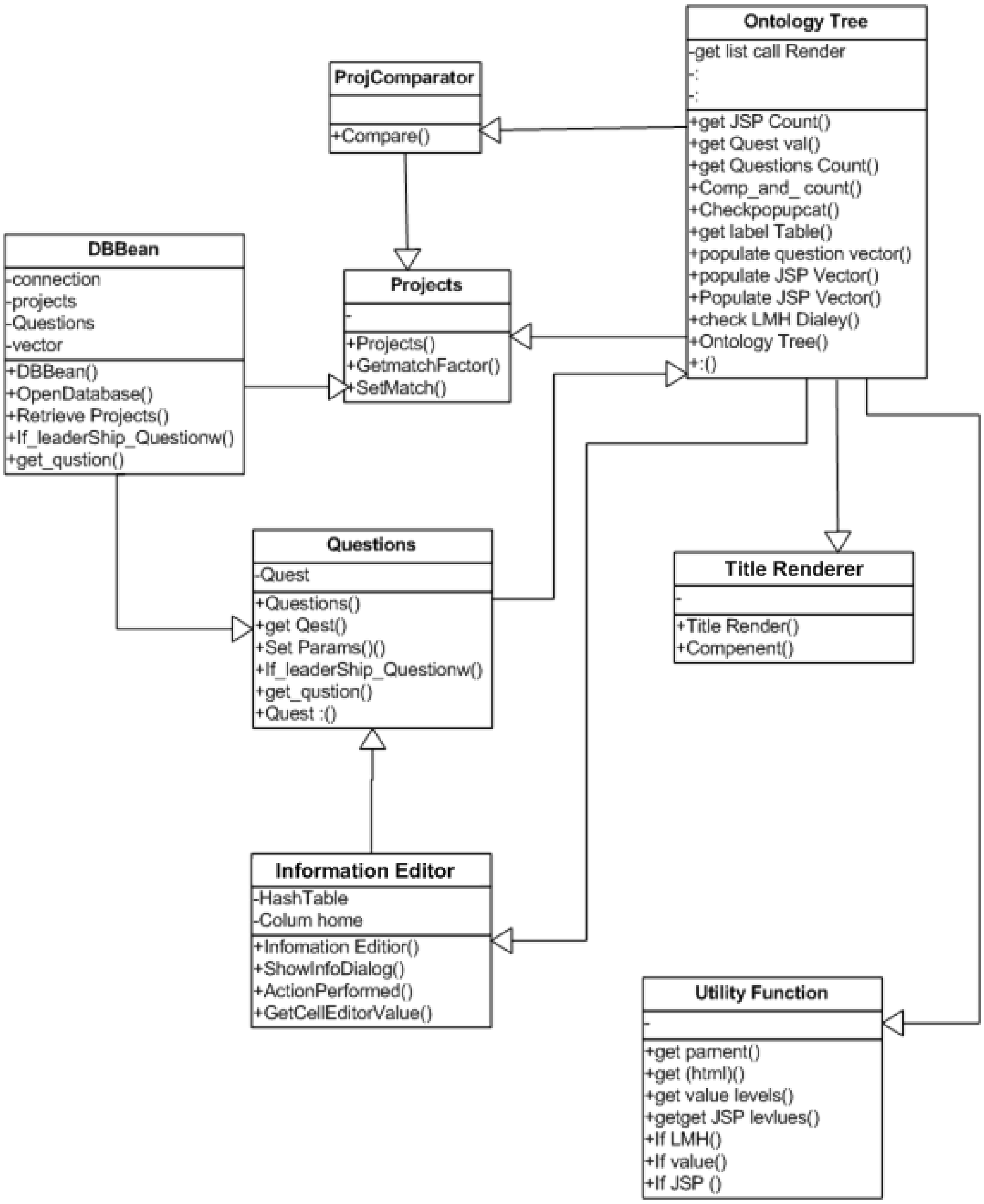


Figure 5.19: SEEOS class diagram

5.10.3 SEEOS Database System

The SEEOS database system consists of 23 tables (see Figure 5.20). These tables contain information about different project attributes. The SEEOS online subsystem contains project information from different organisations and inserts them into the database system.

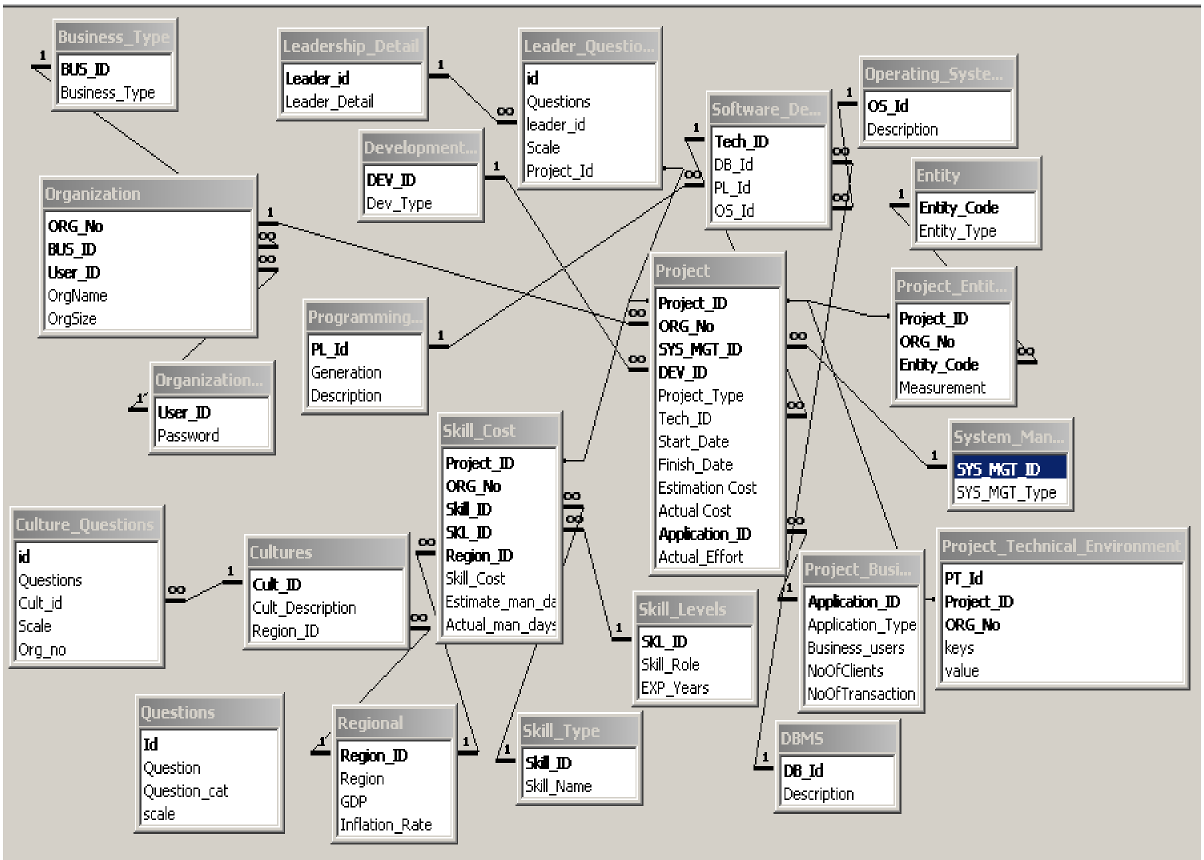


Figure 5.20: Database entities

Each organisation has a user name and password (see Figure 5.21). Each organisation has to input project data such as organisation name, region, organisation type such as public (non-Profit), private (for-profit), and semi-government. Also, the organisation's line of business such as medical, governmental services, telecommunications, tourism services, public services, education and oil and gas has to be added. Other essential information to be added includes organisation types such as 'project-oriented' where the project manager has the highest power in making decisions, 'matrix' where the project manager has moderate power in making decisions, and 'functional' where the project manager has the lowest level of power in making decisions. Of equal importance is the SEEOS user interface, as well as the information about the developed project such as the project name, application types (core or support), project duration, the estimated and actual project cost, the estimated and actual effort (man-days), entities such as number of transactions, application specific information (source line of code, process, objects or class diagrams, tables and entities, technology used, application architecture, skill sets and accumulated years of experience), the project's leadership, and project team cultural characteristics defined with scale point values (see Figure 5.22).

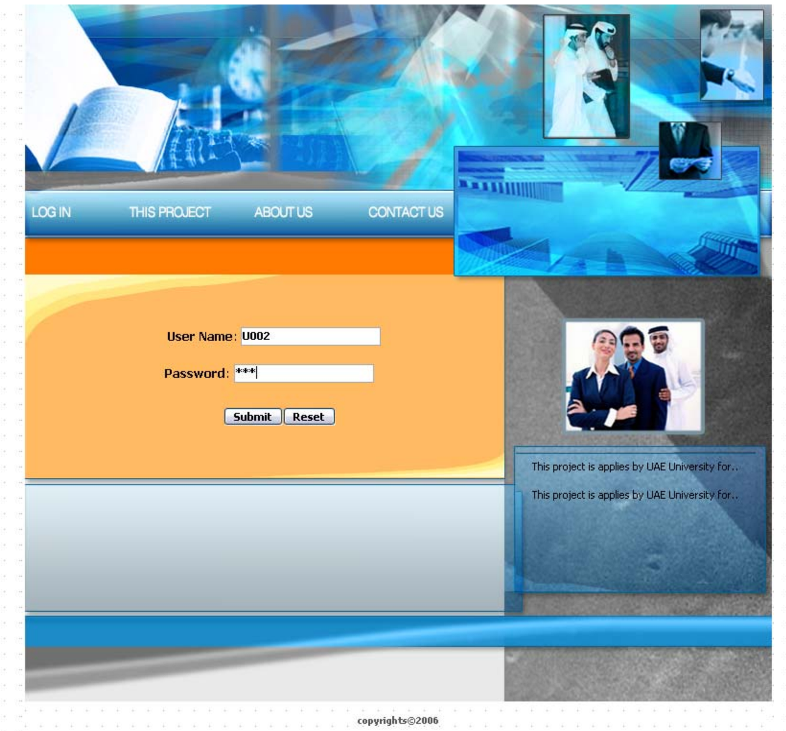


Figure 5.21: Online system login screen

The online system was implemented using various technologies and software such as Adobe Photoshop 7.0 ME to design the interface. Additionally, Java Server Page (JSP) technology was used to manage the content and the appearance of the system. MySQL DBMS was used to implement the database. The SEEOS data definition and data dictionary are listed in Appendix E. The deployment is supported by NetBeans software version 5.0 which is bundled with Apache Tomcat 5.5. This system was partially funded by the UAE University, 2006.

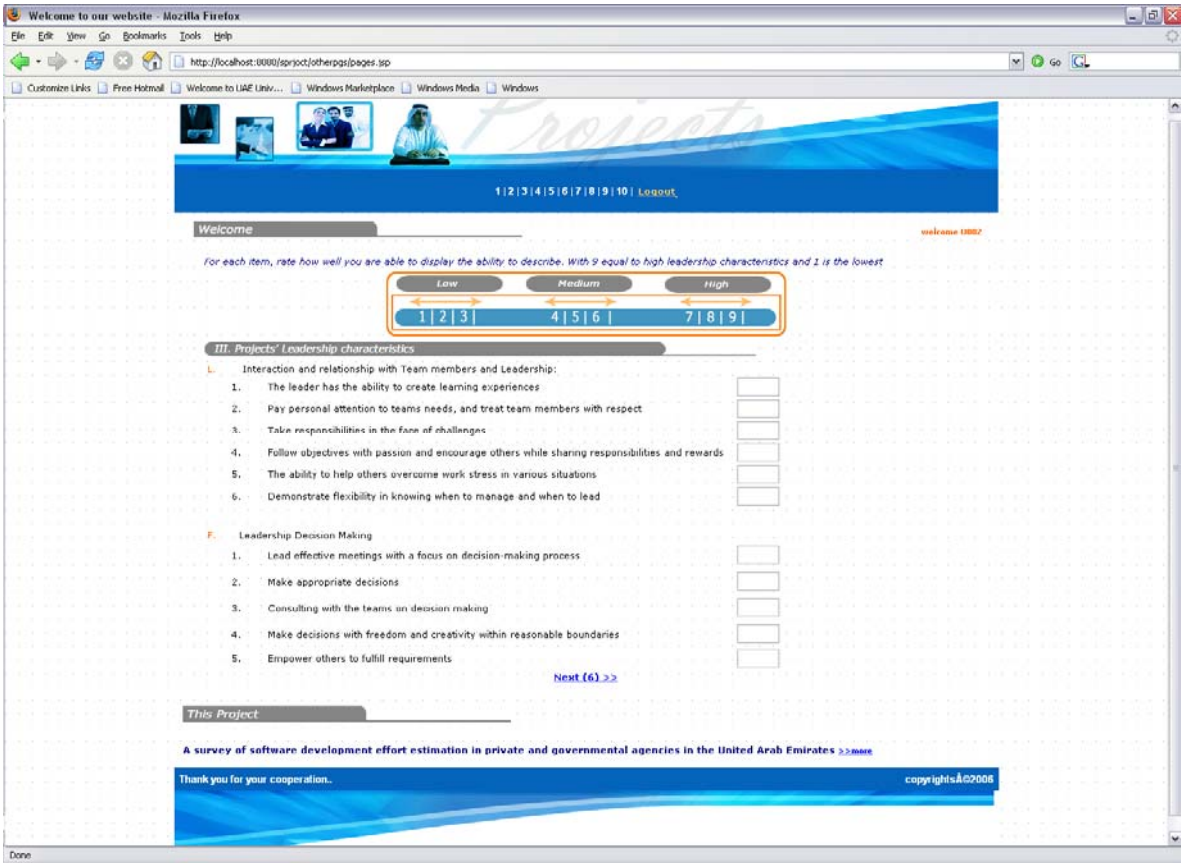


Figure 5.22: Online interface

5.11 SEEOS Validation

The bootstrap subsystem model was developed to enable a simple way of predicting the corrected median effort. This validates the project estimation based on the entry of original analogy effort data, bootstrap sample replication size and effort values. It displays the median bootstrap replications along with the sample median and its standard error. The input interface of the system consists of:

- effort data (number of analogy),
- bootstrap sample replication size value,
- effort values.

The system outputs are replicates of data, mean, median and bias (see Figure 5.23).

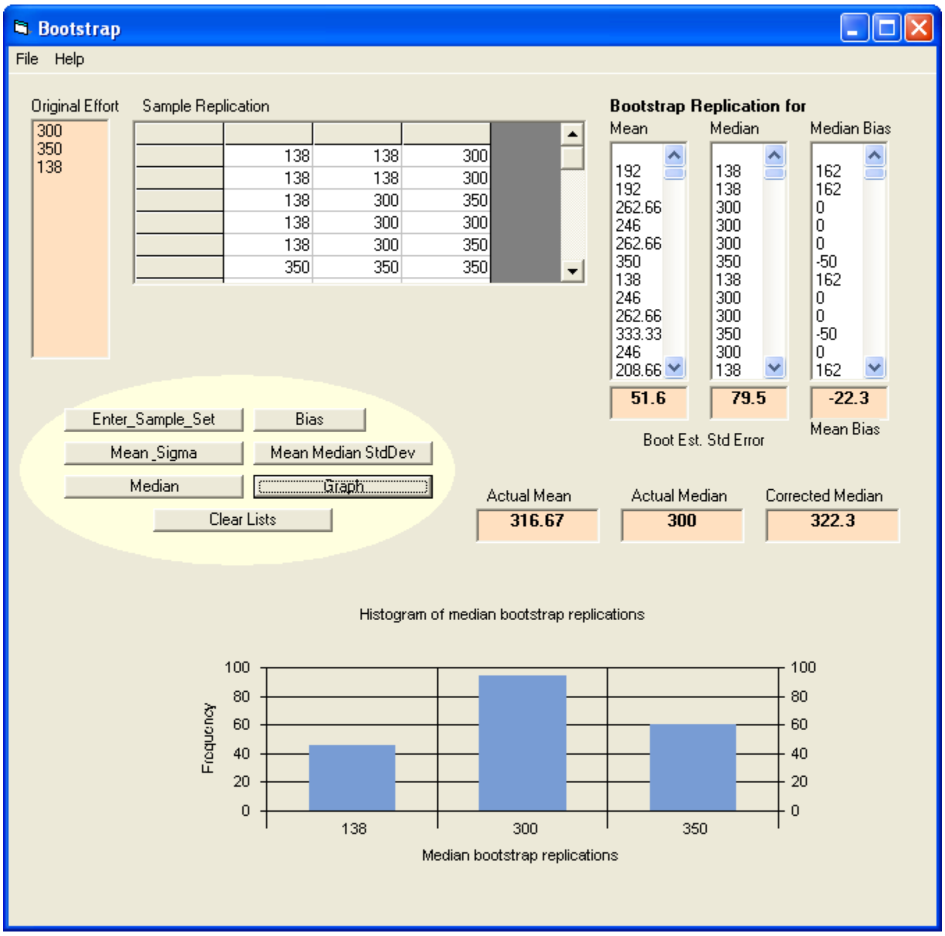


Figure 5.23: Bootstrap user interface

5.12 Summary

This chapter introduced the proposed system architecture and implementation of the augmented CBR model. The system uses analogy methods and addresses issues related to the accuracy of effort estimation by devising a new model. The model includes cultural and leadership characteristics as well as the project manager's authority as attributes. These additions increase the accuracy of effort estimation.

The proposed ontology, as well as the culture and leadership parameters, is captured by this model. The behaviour of the model is consistent with that of other models without the inclusion of culture and leadership. Inclusion of these factors shows a difference and results in more accurate estimations, thus enhancing CBR. This chapter also provides an overview of the major components of the SEEOS. This system consists of three subsystems: the analogy subsystem to find the most similar project, the online subsystem used by different organisations to input project data, and the bootstrap subsystem to validate the project result. The system was implemented using Java and a relational database management system. An online subsystem was developed to populate the database with projects attributes gathered from different organisations within the UAE. The model was used and evaluated by several different organisations and was deemed to be a success.

CHAPTER 6

Results and Evaluation of the Model

6.1 Introduction

The overall concept of the cost estimation model is innovative and contains valuable assets for both public and private sector organisations. The cost estimation model was shown to particular organisations, both government and semi-government and their views were recorded. The general opinion was that the model can be applied for the needs of their organisations. Leaders and managers are keen and enthusiastic to seriously consider and implement more advanced models of software cost estimation. Many organisations were interested in piloting the system due to its applicability to governmental projects. The cost estimation model was also reviewed and examined by IT organisations, for whom it was also found to have potential applications.

This research has considered the importance of culture and leadership in software development projects and particularly in effort estimation. The author has established a new cost estimation model based on Case Based-Reasoning and Ontology-based systems to provide a common understanding of project parameters and semantics, as discussed in Chapters 4 and 5. This new model includes leadership and cultural attributes in an attempt to enhance effort estimation. Appendix D presents statistical analysis of the usefulness of cultural and leadership characteristics in estimating software effort. It also investigates the distribution of cultural and leadership characteristics in UAE-based organisations and their relationship to actual effort and other project attributes, using data from the surveys.

This chapter establishes the importance of leadership and culture in effort estimation. Statistical analysis indicates that effort in core and support systems interact differently with leadership and culture. Moreover, analysis shows that adding leadership and cultural factors to CBR substantially improves the precision of effort estimation.

This chapter specifically presents an evaluation of the cost estimation model, focusing on strengths, weaknesses and potential improvements. An evaluation of the model was carried out to ensure that the development model is useful and reliable in the sense that it can improve cost estimation. An evaluation of the model was made to ensure that it is appropriate and efficient when used in the problem domain of this work. User evaluation of the model was also performed to investigate the usefulness and efficiency of the system. Information was gathered using a system evaluation questionnaire distributed within the selected organisations (see Appendix D).

6.2 Overview of the Evaluation Approach

This chapter presents two methods for estimating actual effort based on regression modeling and Estimation by Analogy (EbA). Separate analyses were generated for core and support system projects due to their different functionalities. This approach was suggested by the analysis in Appendix D which showed that it is difficult to find a valid regression model for estimating actual effort based on all cases. Similarly, this appendix shows that estimation by analogy of the actual effort does not work well using all cases.

Given the importance of total cost as a factor in software development (Hamdan et al., 2005), the estimation of the project's total cost was also considered using both regression modeling and EbA. The accuracy of estimating the actual effort and the total cost using both regression and analogy were assessed using the jack-knife method. The Bootstrap method was used to adjust the bias of the EbA estimate of actual effort. Finally, the actual effort was estimated using Function Point techniques. This included leadership and culture factors to improve the measurement of the project size.

The subsequent sections discuss which aspects of evaluation are covered and how these will be resolved by predicting actual effort and total skill cost using regression and analogy. This chapter is organised as follows: Section 6.4 presents the estimation of actual effort using regression and EbA and assessment of the accuracy of estimates using the jack-knife method. Section 6.5 describes the use of Bootstrap method to adjust for bias the estimate of actual effort, and the overall prediction is displayed. Section 6.6 describes the estimation of the total skill cost, again using both regression and EbA and assessment of the accuracy of estimates and closes with a conclusion in Section 6.7.

The effort estimation model is appropriate for use by most organisations, as was apparent from the research carried out in the various organisations that were involved in this development process. In addition to the general cost estimation model, the team culture and leadership characteristics have also been incorporated to obtain an estimation of the effort of software projects.

6.3 New factors in the SEEOS system

An evaluation of the whole system, including its cultural and leadership attributes, was carried out to investigate how well the SEEOS system performed in comparison to CBR. As a prediction model, it offers the opportunity to project managers or researchers to take corrective action involving all aspects of effort estimation based on the estimated value. The major drawback is that it does not involve the factors related to effort estimation explicitly, but relies on values of the response variable recorded from rare historical data.

6.4 Measuring Project Effort

In the projects considered in this part of the evaluation, government departments developed information systems (applications) to e-enable two types of service: supporting and core systems. Examples of supporting systems are: human resources systems; finance systems; purchasing; document management systems and help desk systems. Supporting services are similar in their functionalities and features across all government departments. Examples of core systems are: fire alarm systems in a civil defence department; flight information systems in a civil aviation department and traffic management systems in police departments.

It is quite difficult to estimate the cost of these services since they have unique functionalities and features. However, in a few cases it was found that some of the functionalities and features in core systems are similar across government departments. For example, all of the above-mentioned systems require a payment module which is similar in those departments. Yet, it was found that 60% of the IT budgets in government departments are spent on supporting services (Hamdan, at el., 2005). The point here is that it would be very reasonable to use a CBR model for software cost estimation for the government departments.

The core systems were detached from supporting systems and treated along with their domain business (lines of business) to see their effect on the effort estimation. A particular problem was the complexity of some of the projects. This often required test data sets to be split.

The validation of the Estimation by Analogy (EbA) model by jack-knife procedure is based on the E_A (actual) and E_E (estimated) values of the total effort amount. Two measures of local error (Table 6.1) are calculated:

1. The magnitude of relative error (MRE) by Conte et al. (1986).
2. The magnitude of relative error to the estimate (MER) by Kitchenham et al. (2001).

$MRE = \frac{\lvert E_A - E_E \rvert}{E_A}$	$MER = \frac{\lvert E_A - E_E \rvert}{E_E}$

Table 6.1: Local accuracy measures

The aforementioned local measures are the basis for the estimation of the global predictive accuracy measures MMRE, $pred_{mre}25$, MMER and $pred_{mer}25$ (see Table 6.2).

$MMRE = \frac{1}{n}\sum_{i=1}^{n}\frac{\lvert E_{Ai} - E_{E_i}\rvert}{E_{Ai}}$	$MMER = \frac{1}{n}\sum_{i=1}^{n}\frac{\lvert E_{Ai} - E_{E_i}\rvert}{E_{E_i}}$
$pred_{mre}25 = \frac{\#(\text{cases with } MRE \leq 0.25)}{\#(\text{cases})}$	$pred_{mer}25 = \frac{\#(\text{cases with } MER \leq 0.25)}{\#(\text{cases})}$

Table 6.2: Global accuracy measures

If the MMRE (or MMER) is small, then these are a good set of predictions. A usual criterion for accepting a prediction method as good is that it has a $MMRE \leq 0.25$ (the same for MMER). The opposite is the case for the $pred_{mre}25$ (or $pred_{mer}25$) accuracy measure. A standard criterion for considering a method as acceptable is $pred_{mre}25 \geq 0.75$ (the same for $pred_{mer}25$).

Next, two methods for predicting effort are presented. The first is based on regression modelling and the second method is based on analogy.

6.4.1 Predicting Effort using Regression

Linear regression was used to evaluate the SEEOS model. The actual effort is a dependent variable with one or more predictors (culture and leadership). The actual effort along with cultural and leadership variables was transformed by a logarithmic function to reduce variability. Moreover, the distribution of the data in the logarithmic scale is closer to the normal distribution than is the actual effort in man-days; this is essential since the linear regression model works better with normal variables. Many important features need close scrutiny in order to predict the cost of a software project. Effort and duration are among many features that require such in-depth study. Usually, the regression models give good results, but the variables have to be transformed first.

The dataset was split according to application type: “Supporting” applications are the systems which support the internal (shared) services in any organisation. These applications are not linked directly to the organisation mission and vision, rather they enhance the efficiency, effectiveness and the performance of the supporting services. Those systems share similar features across the government departments.

“Core” applications exist to help to achieve the mission and vision of the organisations and to satisfy their core purpose. The features of these applications are unique. Organisations with a similar line of business could share similar features. The regression equation for the core applications shows improvement in actual effort, built on 19 cases that were studied.

The model summary for support systems shows the strength of the relationship between the cultural and leadership results and the dependent variable with a multiple correlation coefficient $r = 0.94$ (see Table 6.3). The model explains 75.6% of the variability of the dependent variable LNEffort (p-value <0.007), while the coefficients are presented in Table 6.4. The histogram and the P-P plot of the standardised residuals (see Figure 6.1), show a close to normal distribution. Shapiro-Wilk test of normality on unstandardized residuals (Pallant, 2005) show a (p-value >0.62), which in turn, shows validity of normality assumption.

After attempting regressions with different predictors of subsets, including projects' leadership and team culture attributes, the following predictors were selected for support systems:

- Development tools (programming languages)
- System analyst skills
- Organisational type
- Interaction and relationships
- Ability to motivate
- Communication skills
- Collaboration (interpersonal relationships)
- Reward mechanisms
- Communications
- Team experience

The final model for support systems is:

```
LNEffort_Actual = 7.55 - 0.400 Tools_Language
                       + 0.357 Programming_System_Analyst
                       + 0.455 OrgType - 0.723 Interaction + 0.537 Motivation
                       + 0.901 Communicationskill - 0.830 ImpersonalRelation
                       - 0.741 RewardMechanism + 0.706 Communications
                       - 0.257 TeamExperience
```

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.944[a]	.891	.756	.3500

a. Predictors: (Constant), TeamExperience, Interaction, OrgType, Programming_System_Analyst, Tools_Langua Communications, Motivation, Communicationskill, RewardMechanism, ImpersonalRelation

b. Dependent Variable: LNEffort_Actual

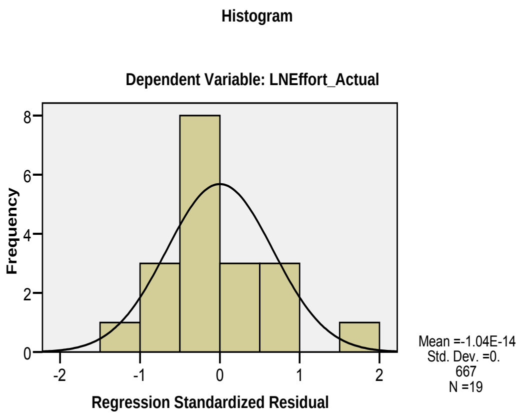
Table 6.3: Model summary (support systems)

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients		
		B	Std. Error	Beta	t	Sig.
1	(Constant)	7.548	.886		8.523	.000
	Tools_Language	-.400	.095	-.563	-4.195	.003
	Programming_System_Analyst	.357	.114	.517	3.137	.014
	OrgType	.455	.177	.413	2.564	.033
	Interaction	-.723	.152	-.975	-4.751	.001
	Motivation	.537	.147	1.068	3.647	.007
	Communicationskill	.901	.192	1.228	4.691	.002
	ImpersonalRelation	-.830	.207	-1.462	-4.006	.004
	RewardMechanism	-.741	.174	-1.301	-4.263	.003
	Communications	.706	.149	1.238	4.747	.001
	TeamExperience	-.257	.105	-.407	-2.438	.041

a. Dependent Variable: LNEffort_Actual

Table 6.4: Coefficients of the linear model (support systems)



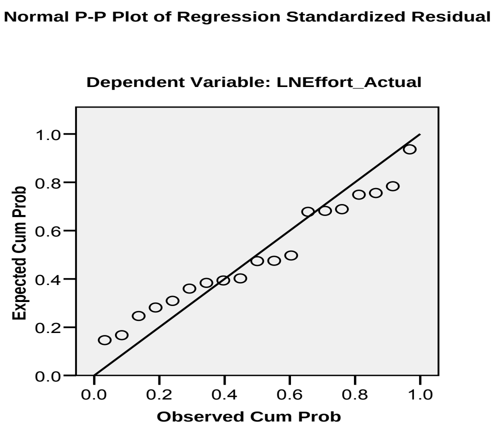


Figure 6.1: Histogram and P-P plot for the Standardized Residuals (support systems)

A multiple regression analysis was performed for the core applications. The model is summarised in Table 6.5. The model explains 72% of variability of the dependent variable actual effort (p-value <0.002), while the coefficients are presented in Table 6.6. The histogram and the P-P plot of the standardized residuals (see Figure 6.2), show a close to normal distribution. The Shapiro-Wilk test of normality on unstandardized residuals shows a (p-value >0.36), which in turn, shows validity of normality assumption.

The following predictors were selected for core systems:

- Organisation's line of business
- Total skills (accumulated experience)
- Application architecture
- Ability to motivate
- Active thinking
- Collaboration (interpersonal relationships)
- Development tools (programming languages)

The final model for core systems is:

```
LNEffort_Actual = 3.19 - 0.214 OrgLOB - 0.0537 Total_Skills
                       + 0.612 App.Architecture + 0.452 Motivation
                       - 0.715 ActiveThinking + 0.596 ImpersonalRelation
                       + 0.302 Tools_Language
```

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.911[a]	.830	.721	.41988

a. Predictors: (Constant), ImpersonalRelation, Tools_Language, Total_Skills, App.Architecture, OrgLOB, Motivation, ActiveThinking

Table 6.5: Model summary (core systems)

Coefficients[a]

Mode		Unstandardized Coefficients B	Std. Erro	Standardized Coefficients Beta	t	Sig.
1	(Constant)	3.188	.923		3.453	.005
	OrgLOB	-.214	.059	-.668	-3.608	.004
	Total_Skills	-.054	.017	-.454	-3.175	.009
	App.Architectu	.612	.177	.516	3.457	.005
	Tools_Langua	.302	.148	.368	2.034	.067
	Motivation	.452	.150	.766	3.025	.012
	ActiveThinking	-.715	.156	-1.335	-4.596	.001
	ImpersonalRel	.596	.196	1.041	3.044	.011

a. Dependent Variable: LNEffort_Actual

Table 6.6: Coefficients of the linear model (core systems)

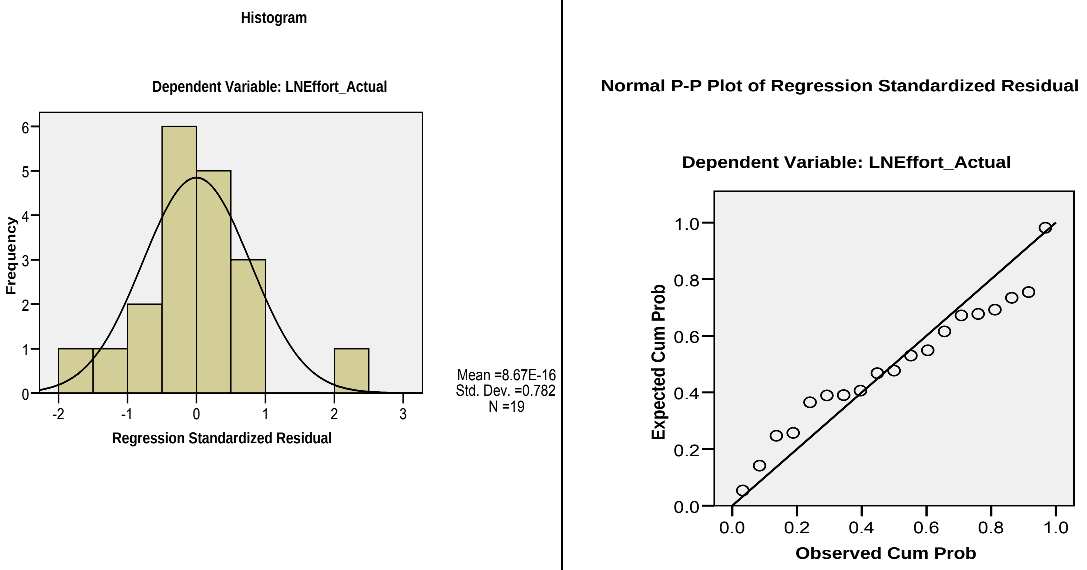


Figure 6.2: Histogram and P-P plot for the Standardized Residuals (core systems)

6.4.2 Predicting Effort using Analogy

In order to estimate the dependent variable effort, the k-nearest neighbour (k-NN) method was used. In particular, various numbers of neighbours in the range 1-10 were tried and the nearest neighbour algorithm by (Aha, 1991) was adopted; the unweighted Euclidean distance measure is the most popular and straightforward distance measure that has been previously used with encouraging results in software engineering cost estimation studies (Angelis and Stamelos, 2000; Shepperd and Schofield, 1997). When there are mixed (numerical and categorical) data, a dissimilarity measure can be computed. An advantage of this measure is that it takes into account the missing values of the projects as well. So, it is believed that this is the most appropriate for missing data. In such a case, the distances are usually calculated using only the available attributes. The measure is suitable for handling those features which are categorical (i.e. nominal or ordinal) or continuous (i.e. interval, ratio or absolute). In this research, the similarities and differences between the different projects' features and the source case that is nearest the target were identified by measuring the distance between cases. The measure was adapted from Shepperd and Schofield, (1997) as shown in equations 1 and 2.

Let $\mathrm{x_{ij}}$ and $\mathrm{y_j}$ *represent* the $\mathrm{j^{th}}$ attributes of cases i and the new case

where

$$d_{new,i} = \sum_{j=1}^{k} w_j^2 \delta\left(Y_j - X_{ij}\right)^{1/2} \qquad (1)$$

$$\delta(Y_j - X_{ij}) = \begin{cases} (\mathrm{y_j} - \mathrm{x_{ij}})^2 & \text{if } \mathrm{y_j} \text{ and } \mathrm{x_{ij}} \text{ are numeric} \\ 0 & \text{if } \mathrm{y_j} = \mathrm{x_{ij}} \text{ and } \mathrm{y_j}, \mathrm{x_{ij}} \text{ are categorical} \\ 1 & \text{if } \mathrm{y_j} \neq \mathrm{x_{ij}} \text{ and } \mathrm{y_j}, \mathrm{x_{ij}} \text{ are categorical} \end{cases} \qquad (2)$$

and $\mathrm{w_j}$ is an appropriate weight

The following similarity measure was used to extract the most similar cases to the new one (a project other than the 38 used as historical cases). For example, if we apply this to case C32 with Y being the new case (on the 30 attributes), the similarity is:

$$sim(32, Y, 30) = \frac{1}{\sqrt{d_{new,32}}} = \frac{1}{\sqrt{1 + 1 + 0 + 1..(6-6)^2 + ..}} \qquad (3)$$

The unknown effort (of the new case) is estimated by a location statistic (mean, median) of those "neighbour" cases, which in this case are the nine values obtained. Here, the median, 600, was used. Regarding the evaluation of the predictive accuracy for the EbA method, the jack-knife procedure was adopted (Kohavi, 1995).

In order to select the appropriate number of analogies the jack-knife technique was applied from one up to ten analogies and the MMRE, MRE, and pred25 accuracy measures were calculated for each of the cases in the whole dataset. It was decided to use one analogy for the predictions, i.e. a number that minimised the MMRE and gave relatively reasonable results for the measures. The values of the effort for the selected cases were: 320, 105, 138, 324, 600, 750, 1250, 1295, and 1300. It appeared that 9 neighbours were a good choice for the construction of EbA model.

Assume (y) is a new case with actual effort 582. First of all, take the absolute value of (actual – estimate) / actual. After applying analogy to estimate the last project (jack-knife y), case 1 is found to be the most similar and 320 is the estimate. However, the true value is 582. So, the relative error for analogy is abs((320 – 582) / 582). The MER will be calculated based on the procedure (actual – estimate) / estimate. So, the MER is abs((582 – 320) / 320) for the first project (see Table 6.7).

OrgNo	Project	OrgSize	OrgLOB	OrgType	Duration	Tools_Language	DBMS	DecisionM	Communicat	TeamExpe	Both C/L	Actual Effort	MRE	MER
29	Telematics s	60	4	2	16	1	2	7	8	6.0	1.00	320	45.02%	81.88%
28	E-Archive	4,000	2	3	11	1	2	8	8	6.0	0.50	1,300	123.37%	55.23%
38	Well Prognos	2,296	7	2	7	3	3	8	8	6.8	0.48	600	3.09%	3.00%
4	Fuder (Data	1,200	7	1	53	5	2	8	8	8.0	0.44	138	76.29%	321.74%
34	Al Ain Munci	2,000	2	1	12	1	1	7	6	6.0	0.42	1,250	114.78%	53.44%
40	Daman Insur	2,296	7	2	4	2	2	8	8	6.8	0.40	324	44.33%	79.63%
30	Project Bus.	200	8	1	24	2	1	7	8	6.0	0.40	1,295	122.51%	55.06%
15	Database Ap	16,000	2	2	6	2	2	7	8	7.0	0.39	750	28.87%	22.40%
1	Financial (JD	1,900	7	2	5	3	3	8	9	8.0	0.37	105	81.96%	454.29%
24	ERP	1,500	2	2	7	1	1	8	8	7.5	0.37	1,200	106.19%	51.50%
32	Planning and	167	2	1	16	1	1	7.5	7.5	6.0		582		

Table 6.7: The dataset example

Three separate groups of independent variables were involved in the procedures. The models that were tried, included, in addition to the other attributes were:

$$effort = f(\text{Project's leadership characteristics, other attributes})$$
$$effort = f(\text{Project team culture, other attributes})$$
$$effort = f(\text{Both groups of characteristics})$$

The analogy was applied to the original data, to some and to all of the variables while effort was a dependent variable. Table 6.8 below shows the results of EbA models on the datasets. The accuracy measures derived from the jack-knife procedure are higher than those derived from regression models but better than those derived from variables when cultural and leadership were not included or when only cultural and leadership were present.

All variables for all cases						
n	MMRE	MMER	MdMRE	MdMER	Pred25mre	Pred25mer
1	99.3%	115.7%	73.6%	67.7%	15.8%	15.8%
2	109.5%	113.1%	69.7%	67.6%	18.4%	17.1%
3	133.9%	121.3%	70.7%	68.5%	16.7%	15.8%
4	132.8%	120.5%	71.4%	67.3%	18.4%	17.1%
5	135.3%	132.4%	71.5%	69.0%	17.9%	16.8%
6	138.1%	135.1%	71.7%	71.1%	17.1%	17.1%
7	137.0%	131.8%	71.7%	70.2%	18.0%	17.3%
8	136.3%	134.0%	71.5%	70.0%	18.1%	17.8%
9	131.5%	130.4%	70.8%	68.5%	18.7%	19.0%
10	130.7%	129.5%	69.9%	69.2%	18.4%	18.4%
All variables without Culture and Leadership						
n	MMRE	MMER	MdMRE	MdMER	Pred25mre	Pred25mer
1	94.3%	114.2%	55.2%	62.2%	28.9%	28.9%
2	109.9%	138.3%	67.9%	67.6%	22.4%	21.1%
3	103.6%	119.6%	65.8%	59.1%	22.8%	22.8%
4	133.9%	117.8%	67.6%	66.6%	23.0%	21.1%
5	132.1%	119.9%	68.9%	67.6%	20.0%	18.9%
6	136.7%	124.1%	70.0%	67.6%	19.3%	18.0%
7	138.9%	125.8%	70.8%	69.0%	18.4%	17.7%
8	137.4%	125.3%	71.8%	67.6%	17.4%	16.8%
9	139.0%	121.9%	71.4%	67.6%	17.5%	17.3%
10	136.3%	122.4%	71.2%	67.6%	17.6%	17.4%
Project Team Culture only						
n	MMRE	MMER	MdMRE	MdMER	Pred25mre	Pred25mer
1	136.9%	117.4%	66.8%	78.5%	15.8%	18.4%
2	133.1%	140.8%	67.4%	66.9%	14.5%	14.5%
3	126.1%	146.4%	69.2%	69.5%	14.9%	14.0%
4	125.8%	146.0%	66.6%	70.3%	16.4%	16.4%
5	135.8%	141.3%	70.7%	70.9%	16.3%	16.3%
6	136.1%	132.3%	68.8%	69.5%	18.0%	18.4%
7	131.8%	123.1%	65.1%	67.4%	19.2%	19.5%
8	154.3%	141.8%	68.8%	67.6%	18.4%	18.4%
9	132.7%	124.1%	67.4%	67.6%	18.4%	18.7%
10	137.2%	119.1%	67.0%	67.2%	19.2%	20.0%
Project's Leadership characterstics only						
n	MMRE	MMER	MdMRE	MdMER	Pred25mre	Pred25mer
1	115.8%	97.1%	60.4%	68.8%	26.3%	26.3%
2	98.6%	111.5%	61.7%	72.0%	19.7%	19.7%
3	107.8%	104.4%	65.4%	71.1%	20.2%	20.2%
4	124.1%	114.3%	69.1%	70.8%	19.7%	19.1%
5	116.2%	116.6%	68.9%	68.9%	20.5%	20.0%
6	116.9%	118.0%	67.0%	69.1%	20.2%	20.6%
7	118.1%	119.1%	68.5%	69.8%	18.8%	18.8%
8	142.5%	142.1%	68.5%	68.7%	19.4%	19.4%
9	120.0%	120.4%	68.1%	69.6%	19.6%	19.0%
10	123.3%	123.8%	69.2%	71.1%	18.7%	18.4%
Both Culture and Leadership only						
n	MMRE	MMER	MdMRE	MdMER	Pred25mre	Pred25mer
1	134.4%	96.8%	55.7%	67.4%	28.9%	31.6%
2	122.3%	95.0%	57.6%	69.9%	26.3%	26.3%
3	124.3%	116.5%	61.6%	72.8%	22.8%	21.1%
4	123.8%	126.8%	68.7%	72.8%	19.7%	19.1%
5	123.2%	122.8%	70.0%	70.5%	19.5%	19.5%
6	125.2%	120.1%	70.0%	70.5%	19.7%	19.7%
7	126.6%	121.6%	69.2%	71.1%	19.2%	19.2%
8	154.2%	151.3%	71.1%	71.9%	18.4%	18.4%
9	130.2%	130.1%	69.2%	71.7%	18.7%	19.0%
10	131.5%	131.6%	70.2%	71.7%	19.2%	18.9%

Table 6.8: Effort accuracy measures on all cases

Next, the (core-support) models were intended to measure predictive accuracy (MMRE, Pred25) with and without cultural and leadership characteristics on the split cases. The functionalities of these systems are different and should be treated separately. The analogy indeed showed significant differences between cases for the support systems with 90 percent of the cases including cultural and leadership characteristics which improved the analogy. The core applications improved the analogy by 50 percent when the two highest effort cases were removed (see Table 6.9).

There are 19 core projects and 17 support projects in which there are no missing values for the dependent variable.

Core Systems							Support Systems						
All variables							All variables						
n	MMRE	MMER	MdMRE	MdMER	Pred25mre	Pred25mer	n	MMRE	MMER	MdMRE	MdMER	Pred25mre	Pred25mer
1	106.0%	111.6%	70.9%	67.6%	15.8%	15.8%	1	225.8%	163.2%	86.0%	80.8%	5.3%	0.0%
2	129.1%	119.0%	75.5%	72.4%	13.2%	13.2%	2	170.4%	133.0%	78.9%	73.0%	10.5%	7.9%
3	111.9%	122.2%	70.9%	70.9%	15.8%	15.8%	3	155.3%	128.1%	76.9%	73.0%	15.8%	15.8%
4	113.5%	114.2%	69.1%	70.1%	18.4%	19.7%	4	144.5%	149.7%	74.1%	73.0%	15.8%	17.1%
5	107.1%	111.0%	67.6%	69.2%	17.9%	18.9%	5	143.5%	138.5%	73.0%	72.7%	15.8%	16.8%
6	110.1%	112.9%	63.5%	68.8%	18.4%	19.3%	6	142.9%	130.8%	71.5%	70.8%	16.7%	16.7%
7	108.2%	110.8%	63.7%	68.3%	20.3%	20.3%	7	131.5%	122.0%	70.0%	67.9%	17.3%	18.0%
8	105.0%	106.2%	62.9%	65.4%	21.7%	21.1%	8	129.4%	120.2%	69.4%	68.8%	16.4%	17.1%
9	102.4%	101.8%	62.4%	61.9%	22.2%	21.6%	9	132.6%	120.7%	68.8%	68.8%	17.0%	17.5%
10	102.1%	104.6%	62.7%	60.9%	21.6%	20.5%	10	132.6%	123.4%	68.8%	69.4%	17.4%	16.8%
All variables without Culture and Leadership							All variables without Culture and Leadership						
n	MMRE	MMER	MdMRE	MdMER	Pred25mre	Pred25mer	n	MMRE	MMER	MdMRE	MdMER	Pred25mre	Pred25mer
1	74.8%	125.7%	70.9%	56.9%	21.1%	21.1%	1	162.8%	200.1%	80.8%	100.0%	10.5%	10.5%
2	101.6%	115.1%	68.8%	68.8%	18.4%	18.4%	2	185.8%	149.2%	77.0%	81.9%	18.4%	15.8%
3	125.8%	127.4%	78.3%	70.9%	17.5%	17.5%	3	170.8%	139.0%	79.1%	75.8%	14.0%	10.5%
4	110.4%	112.1%	69.1%	67.9%	19.7%	21.1%	4	160.4%	143.1%	76.4%	75.1%	14.5%	13.2%
5	110.1%	106.3%	68.3%	67.6%	20.0%	21.1%	5	157.2%	128.4%	74.0%	73.0%	13.7%	12.6%
6	112.6%	104.3%	64.4%	67.9%	21.9%	23.7%	6	142.1%	120.5%	72.3%	69.4%	15.8%	14.9%
7	105.9%	106.9%	62.4%	65.2%	24.1%	24.8%	7	135.8%	116.6%	70.7%	68.8%	15.8%	16.5%
8	104.8%	101.3%	62.4%	60.9%	23.7%	23.7%	8	137.9%	125.2%	70.3%	67.4%	16.4%	16.4%
9	104.4%	100.2%	61.9%	62.4%	22.8%	21.6%	9	137.6%	118.6%	68.8%	64.6%	17.5%	17.0%
10	104.4%	104.3%	62.9%	62.1%	21.6%	20.0%	10	129.5%	118.5%	65.8%	64.0%	18.4%	17.9%
Culture and Leadership Variables (Both)							Both Culture and Leadership Variables (Both)						
n	MMRE	MMER	MdMRE	MdMER	Pred25mre	Pred25mer	n	MMRE	MMER	MdMRE	MdMER	Pred25mre	Pred25mer
1	115.2%	95.4%	60.9%	73.9%	15.8%	15.8%	1	118.3%	157.8%	74.4%	73.1%	21.1%	21.1%
2	128.7%	87.8%	68.3%	67.7%	13.2%	13.2%	2	166.7%	163.6%	73.9%	77.0%	10.5%	13.2%
3	126.7%	91.9%	68.3%	63.7%	14.0%	15.8%	3	141.1%	149.3%	73.0%	77.1%	14.0%	14.0%
4	122.5%	101.1%	69.1%	67.7%	13.2%	14.5%	4	131.2%	152.7%	74.4%	73.2%	15.8%	17.1%
5	115.6%	95.1%	69.2%	63.0%	14.7%	15.8%	5	141.5%	150.1%	74.3%	73.1%	15.8%	16.8%
6	114.5%	103.4%	69.1%	63.5%	14.0%	14.9%	6	137.8%	150.1%	73.0%	73.9%	14.9%	15.8%
7	109.4%	102.0%	68.3%	61.9%	14.3%	15.8%	7	134.5%	140.8%	72.7%	73.0%	15.0%	15.8%
8	117.4%	103.9%	66.4%	62.6%	14.5%	15.8%	8	130.8%	133.0%	71.9%	71.0%	16.4%	17.1%
9	118.0%	112.2%	67.2%	63.3%	15.2%	14.6%	9	125.5%	128.6%	70.0%	70.0%	17.5%	18.1%
10	110.1%	112.8%	62.3%	65.4%	16.8%	15.8%	10	129.9%	129.4%	70.0%	70.0%	16.8%	16.8%

Table 6.9: Effort accuracy measures for core and support systems

In the presence of correlated independent variables, the regression coefficient may not be meaningful. The negative coefficients in equations do not reflect the true effects of independent variables. The fitting accuracy of the model is presented in Table 6.10 and Table 6.11. In order to evaluate the predictive accuracy, the jack-knife procedure was used. Then two different MMRE were calculated:

1) The "fitting" MMRE: this is calculated by the regression procedure in SPSS. The "Unstandardized" predicted values are computed for the data that were used to fit the model and these are in fact the predicted logarithm of the efforts. The computed MREs are therefore given by:

$$\left| \frac{e^{predicted} - e^{LN(effort)}}{e^{LN(effort)}} \right|$$

Next, the mean of all MREs gives the MMRE.

2) The "predictive" MMRE: this is computed when the jack-knife procedure is applied and it can be also be computed in SPSS by the "deleted" residuals. These residuals (say r) are computed as the differences r =ln(effort) - predict; but here the prediction is made for each case when this is deleted from the data. So by computing first: Predicted = ln(effort) - r

The jack-knife MRE is:

$$\text{MRE=} \quad \left| \frac{e^{predicted} - e^{LN(effort)}}{e^{LN(effort)}} \right|$$

The mean and median of all MRE is the predictive MMRE. After calculating both MMRE and MMER, the corresponding pred25 measure for them.

		MMRE	MdMRE	MMER	MdMER	predMRE25	predMER25
Regression	**Fitting Accuracy**	20%	17%	21%	19%	79%	68%
	Predictive Accuracy	56%	47%	57%	45%	16%	26%
EbA best model (n=8)	**Predictive Accuracy**	129 %	120%	69%	68%	16%	17%

Table 6.10: Accuracy measures for the linear regression model (support systems)

		MMRE	MdMRE	MMER	MdMER	predMRE25	predMER25
Regression	**Fitting Accuracy**	24%	17%	25%	21%	74%	74%
	Predictive Accuracy	42%	28%	44%	30%	37%	42%
EbA best model (n=9)	**Predictive Accuracy**	102%	101%	62%	62%	22%	22%

Table 6.11: Accuracy measures for the linear regression model (core systems)

The comparison of the two models shows that the linear regression model outperforms EbA for all support and core systems measures. On the other hand, analysis of the completed projects, including leadership and cultural attributes appears to provide better results. Regression and analogy performed better when cases were split and selected as core and support systems.

6.5 Bootstrap and Bias Techniques

The Bootstrap is a computer-based method for the estimation of standard error (SE), bias, confidence intervals, and other measures of statistical accuracy (Efron and Tibshirani, 1993). The computations show how useful and simple the bootstrap is in estimating the median effort and its standard error. The use of Bootstrap in estimating the effort involved in a given project along with certain methods like analogies and regression using real historical data, is of great interest due to difficulties in describing quantitative concepts such as effort, which are rather abstract when it comes to measuring. On the other hand, these methods require very little in terms of mathematical background and assumptions, but rely on computing power instead. The Bootstrap can calculate standard errors for statistics that theory does not easily handle and is a method which is widely used nowadays. The approach is general, yet simple and instructive. It has found use on the robust estimation, sampling, and a variety of other applications.

For example, an unknown parameter θ = $t(F)$ is estimated by $\hat{\theta}$ = $s(x)$ on the basis of a random sample $x = (x_1, x_2, \ldots\ldots, x_n)$ from the probability distribution F (see Figure 6.3). This shows the procedure used by the Bootstrap to estimate the standard error of s(x).

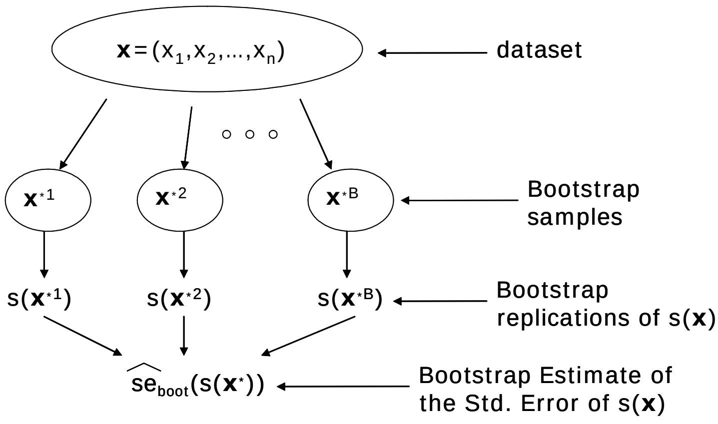


Figure 6.3: Standard Bootstrap diagram

Here $\widehat{se}_{boot}(s(\mathbf{x}^*))$ is the sample standard deviation of $s(x^{*k}), k=1,....,B$. Generally, the Bootstrap gives adequate results for B (# of replications) between 25 and 200 (Efron and Tibshirani, 1986). A bootstrap sample $x^* = (x^{*1}, x^{*2}, ..., x^{*B})$ is a random sample of size n drawn $\hat{F}$. For the Bootstrap SE estimate, it is known that $\hat{F}$ $\widehat{se}_{boot}(s(\mathbf{x}^*)) = se_{\hat{F}}(\hat{\theta}^*)$. Here $\hat{F}$ indicates the empirical distribution, and the limit of the left side is the ideal (but not fully accurate) estimate of the SE of s(x*). In fact, the success of the bootstrapping is partially due to the fact that $\hat{F}$ is consistent for F.

In theory, Bootstrap estimates the standard deviation of the sample median. This is normally given by the standard error of the sample median $s(x)$, i.e. using the distribution of the median which is not obvious. The Bootstrap estimate of the standard error (calculated for the sample median) gives an easy and practical answer. A Minitab macro (Bootstrap) was written to do the sampling and calculations. The median Bootstrap replications were then displayed along with the sample median and the SE of the Bootstrap estimate. The algorithm used for the macro Bootstrap is given below (Bellout, Harbi, and Hamdan, 2006).

```
Declare column variables x, y, md, mdt, seboot, med
Declare constant variables b, szx, i
Count x and store the result in szx
do i = 1 : b
    sample szx elements from x with replacement and store them in y
    Find the median of y and store it in mdt.
    let md(i) = mdt(1)
enddo
Find the standard deviation of md and store it in seboot
Find the median of x and store it in med
print md
print med, seboot
Plot the histogram of md
```

The execution of the macro using Minitab gave: s(x) = 600 and $\widehat{se}_{boot}(s(\mathbf{x}^*))$ = 348. How good is this estimate? Compare it to $se_{\hat{F}}(\hat{\theta}^*)$, which is obtained from the sampling distribution of s(x*) bootstrap replications (see Table 6.12).

	bootstrap	
		estimate
Row	median	of se
1	600	348.742

Table 6.12: The bootstrap estimate

This section computes the standard of the bootstrap median based on all possible bootstrap samples, 9^9 samples.

$$p_i = P\{s(\mathbf{x}^*) = x_{(i)}\} = \sum_{j=0}^{4}\left[Bi\left(j;9;\frac{i-1}{9}\right) - Bi\left(j;9;\frac{i}{9}\right)\right]$$

The results are given in Table 6.13:

x(i)	p(i)
105	0.0014493
138	0.0289240
320	0.1144725
324	0.2206611
600	0.2689862
750	0.2206611
1250	0.1144725
1295	0.0289240
1300	0.0014493

Table 6.13: The sampling distribution of the sample median

The previous distribution gives $se_{\hat{F}}(\hat{\theta}^*) = 349.5$ which is very close to the Bootstrap estimate obtained earlier. To have more insight about these estimates, histograms were constructed of the 200 replications used in the bootstrap estimate and also of 200 observations generated from the previous distribution of s(x*). These turned out to be similar (see Figure 6.4).

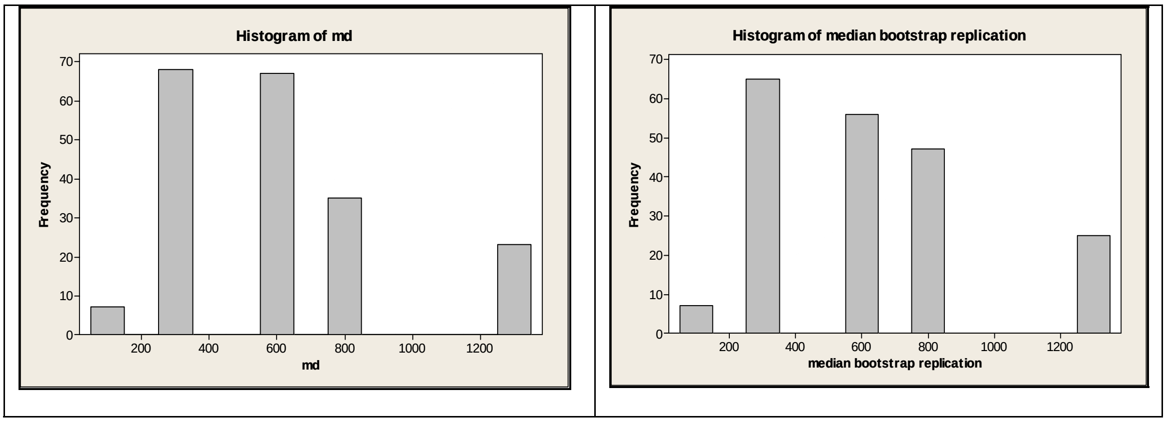


Figure 6.4: Simulated and Bootstrap histograms of the median distribution

The Bootstrap estimate of the median bias was used to estimate the bias of the sample median (which is 600 here, as seen earlier). A Minitab macro (bootstrap bias), was written to display the bias bootstrap replications along with the sample median and its bootstrap estimate of the bias. The execution of the macro bootstrap bias gave $\widehat{bias}_{boot}(s(\mathbf{x}^*)) = 18.5$ which is the bootstrap estimate of the median bias. The Bootstrap estimate of the bias is as shown (see Table 6.14):

bootstrap		
Row	estimate of the median	bias
1	600	18.52

Table 6.14: The bootstrap bias estimate

This positive value suggests that the sample median (of 600) overestimates the actual median, and needs correction. A validation operation is then needed and the corrected median is then s(x) corrected = 600 – $\widehat{bias}_{boot}(s(\mathbf{x}^*))$ = 600 – (18.5). So the estimate (prediction in fact) of the "effort" for the new project is: s(x) corrected = 581.5 (very close to the actual effort 582 and the median effort 600). The corrected median is then 582 person-days.

The Bootstrap provides re-sampling methods that are widely used to estimate parameters and evaluate the biases and the errors of estimation with little or no assumptions about the underlying distribution. When applied to the problem of project effort estimation it gives good results close to the actual effort. This approach is general, yet is simple and instructive. Its success is partially due to its ease of implementation and nonparametric nature. The Bootstrap estimate of the bias in estimating the actual effort is used to adjust and correct estimates of actual efforts.

6.6 Measuring Project Budget: Total Skill Cost

Total cost is computed as total cost for the complete project. This involves each skill set and the amount of work on the project for a period of time (in years). The attribute 'man-year' was calculated by the number of skilled staff and the number of years. In this study, the rate was sourced as in Janulaitis (2005), along with a study of comparative salaries. The cost was a multiplication of rate per month by 12 months and by man-years. This was repeated with other skills. The total skill cost was calculated by adding all costs. The person-days (effort) was calculated in days (see Table 6.15). The days in a year are approximately 185 excluding holidays and could vary in some workplaces.

Skills	No	years	Person-Years	Days/year	Person-Days	rate/M	Cost
Analyst	10	4	40			40000	19,200,000
DB Desn	5	3	15			35000	6,300,000
Network	10	2	20			30000	7,200,000
Project Manager	1	4	4			60000	2,880,000
QA	1	1	1			30000	360,000
Total	27		80	185	14800		$35,940,000

Table 6.15: Total actual project cost for sample project

A model was built with the total skill cost as the dependent variable. The descriptive statistics of the dependent variable are presented in Table 6.16. There are 41 projects with no missing values for the dependent variable. In order to test the null hypothesis that the dependent variable comes from the normal distribution, the One-Sample Kolmogorov-Smirnov procedure (see Table 6.17) and plots were used (see Figure 6.5). There is evidence that the dependent variable does not follow the normality so transformations of the variables are required. Logarithmic transformation was utilised in order to achieve the normality.

Statistics

Total Skill Cost

N	Valid	41
	Missing	0
Mean		2052259
Median		1223040
Std. Deviation		2320646
Skewness		2,470
Std. Error of Skewness		,369
Minimum		160000
Maximum		10535958

Table 6.16: Descriptive statistics for total skill cost

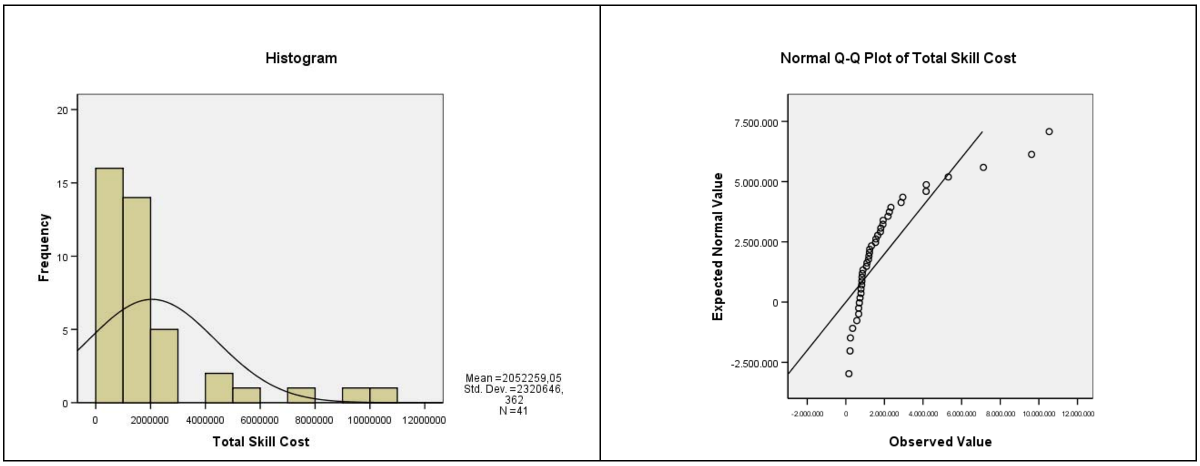


Figure 6.5: Histogram and Q-Q plot for Total Skill Cost

One-Sample Kolmogorov-Smirnov Test

		Total Skill Cost
N		41
Normal Parameters[a,b]	Mean	2052259
	Std. Deviation	2320646
Most Extreme Differences	Absolute	,256
	Positive	,256
	Negative	-,207
Kolmogorov-Smirnov Z		1,641
Asymp. Sig. (2-tailed)		,009

a. Test distribution is Normal.

b. Calculated from data.

Table 6.17: One-Sample Kolmogorov-Smirnov test for total skill cost

The descriptive statistics of the new dependent variable (LNTotalSkillCost) are presented in Table 6.18. The One-Sample Kolmogorov-Smirnov procedure (see Table 6.19) and plots (see Figure 6.6) show that the new variable follows the normal distribution.

Statistics

LNTotalSkillCost

N	Valid	41
	Missing	0
Mean		14,0973
Median		14,0169
Std. Deviation		,93225
Skewness		,115
Std. Error of Skewness		,369
Minimum		11,98
Maximum		16,17

Table 6.18: Descriptive statistics for LNTotalSkillCost

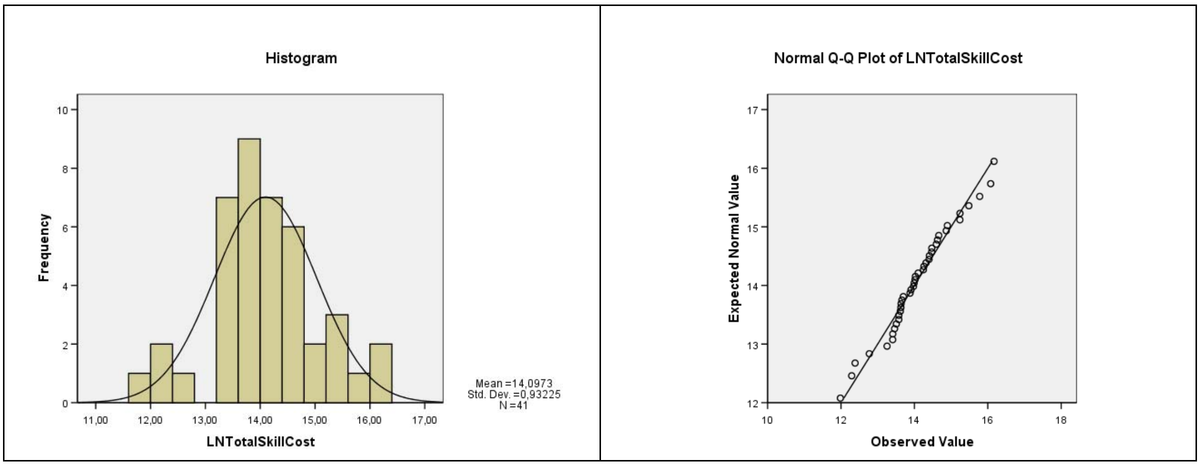


Figure 6.6: Histogram and Q-Q plot for LN Total Skill Cost

One-Sample Kolmogorov-Smirnov Test

		LNTotalSkill Cost
N		41
Normal Parameters[a,b]	Mean	14,0973
	Std. Deviation	,93225
Most Extreme Differences	Absolute	,108
	Positive	,077
	Negative	-,108
Kolmogorov-Smirnov Z		,692
Asymp. Sig. (2-tailed)		,725

a. Test distribution is Normal.

b. Calculated from data.

Table 6.19: One-Sample Kolmogorov-Smirnov test for LNTotalSkillCost

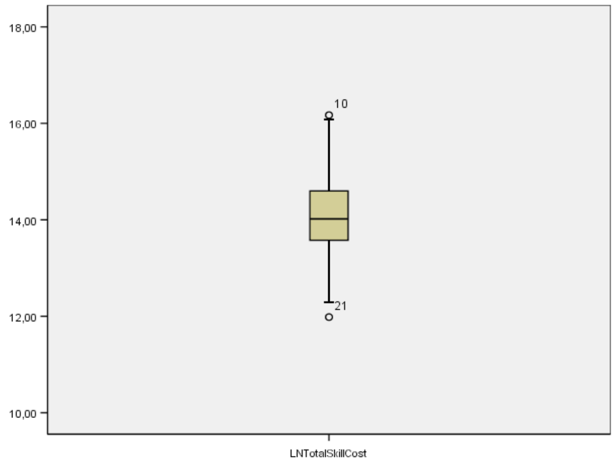


Figure 6.7: Box plot for the LN Total Skill Cost

After applying the one-way ANOVA test for the LNTotalSkillCost and the nominal variables of the dataset, it can be seen that Application Architecture has an impact on the dependent variable (sig<0.056) (Table 6.20).

LNTotalSkillCost

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	4,890	2	2,445	3,110	,056
Within Groups	29,874	38	,786		
Total	34,764	40			

Table 6.20: One-way ANOVA for the LNTotalSkillCost and application architecture

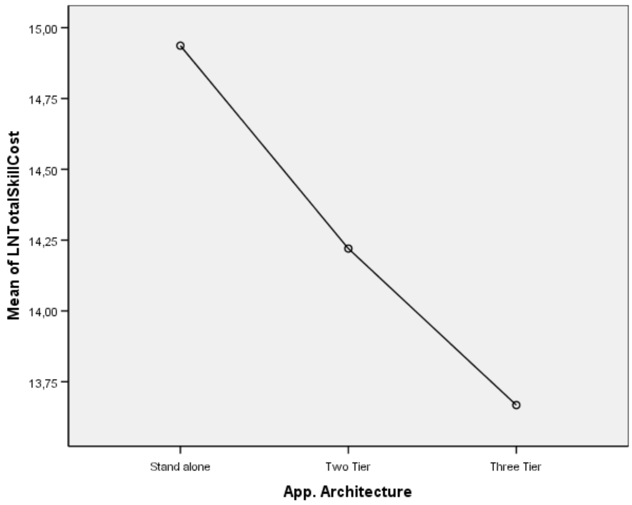


Figure 6.8: Means plot of LN Total Skill Cost for application architecture

6.6.1 Linear Regression: Total Skill Cost

The dependent variable LNTotalSkillCost was used for the construction of a linear regression model. After the entrance and removal of the Project's Leadership characteristics and Project Team Culture independent variables, the following predictors were selected:

Project's Leadership characteristics:

- Interaction and relationships
- Decision-Making
- Communications

Project Team Culture:

- Timeliness
- Job Stability
- Communications

The model explains 48.2 percent (see Table 6.21) of the variability of the dependent variable LNTotalSkillCost (sig<0.000), whereas the coefficients are presented in Table 6.22. In the histogram of the standardized residuals (Figure 6.9), a slight left skewness is highlighted. The final model is:

$$LNTotalSkillCost = 14.848 + 0.532 * Interaction + 0.921 * DecisionMaking - 1.250 * CommunicationSkill - 0.245 * TimeRespect - 0.481 * JobStability + 0.435 * Communications$$

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,748[a]	,559	,482	,67124

a. Predictors: (Constant), Communications, Job Stability, Communication skill, Time - Respect, Interaction, Decision Making

Table 6.21: Model summary (LNTotalSkillCost)

Coefficients[a]

Model		Unstandardized Coefficients B	Unstandardized Coefficients Std. Error	Standardized Coefficients Beta	t	Sig.
1	(Constant)	14,848	,839		17,698	,000
	Interaction	,532	,190	,683	2,795	,008
	Decision Making	,921	,209	1,237	4,406	,000
	Communication skill	-1,250	,265	-1,451	-4,720	,000
	Time - Respect	-,245	,132	-,383	-1,858	,072
	Job Stability	-,481	,176	-,616	-2,736	,010
	Communications	,435	,122	,613	3,574	,001

a. Dependent Variable: LNTotalSkillCost

Table 6.22: Coefficients of the linear model (LNTotalSkillCost)

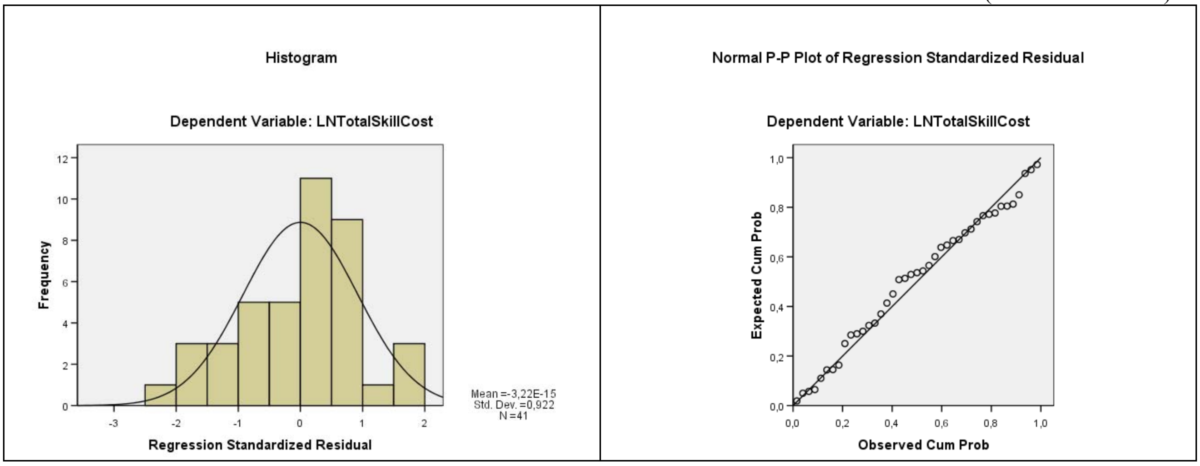


Figure 6.9: Histogram and P-P plot for the Standardized Residuals (LNTotalSkillCost)

The fitting accuracy of the model is presented in Table 6.23. In order to evaluate the predictive accuracy, the jack-knife procedure was used.

After applying linear regression on the project's leadership characteristics and project team culture attributes separately, it was concluded that a representative model for the dependent variable LNTotalSkillCost could not be built.

		MMRE	MdMRE	MMER	MdMER	predMRE25	predMER25
Regression	**Fitting Accuracy**	57.37%	39.32%	53.53%	41.36%	29.00%	24.00%
	Predictive Accuracy	72.07%	43.35%	68.33%	49.37%	21.95%	21.95%
EbA best model (n=7)	**Predictive Accuracy**	79.65%	49.13%	76.55%	52.96%	24.39%	34.15%

Table 6.23: Accuracy measures for the linear regression model (LNTotalSkillCost)

6.6.2 Estimation by Analogy: Total Skill Cost

Estimation by analogy (EbA) is another technique for the prediction of a dependent variable. Various neighbours were tried out and the results of the jack-knife procedure are presented in Table 6.24. As observed, the optimal number of neighbours varies according to the accuracy measure that needs to be optimised. It would appear that 7 neighbours is a good choice for the construction of an EbA model.

No of Neighbors	MMRE	MdMRE	MMER	MdMER	predMRE25	predMER25
1	101.21%	67.34%	168.48%	63.68%	17.07%	17.07%
2	86.26%	52.34%	95.37%	52.99%	31.71%	26.83%
3	87.44%	55.17%	96.74%	58.34%	24.39%	21.95%
4	80.52%	56.65%	87.55%	53.40%	24.39%	24.39%
5	82.16%	55.41%	82.12%	57.97%	17.07%	19.51%
6	84.63%	54.43%	80.54%	62.73%	26.83%	24.39%
7	79.65%	49.13%	76.55%	52.96%	24.39%	34.15%
8	83.03%	53.23%	78.60%	52.89%	24.39%	24.39%
9	90.94%	58.60%	78.09%	49.24%	19.51%	21.95%
10	101.20%	58.33%	74.24%	45.16%	24.39%	21.95%
11	125.62%	64.80%	78.99%	52.94%	19.51%	21.95%
12	137.85%	78.91%	81.32%	55.49%	19.51%	21.95%
13	141.94%	79.94%	80.95%	55.19%	21.95%	19.51%
14	149.54%	80.75%	82.68%	56.84%	17.07%	17.07%
15	147.01%	80.36%	80.06%	56.82%	14.63%	19.51%
16	144.00%	73.10%	78.36%	55.52%	17.07%	19.51%
17	148.19%	71.72%	79.45%	56.80%	14.63%	17.07%
18	159.23%	73.38%	78.28%	60.85%	12.20%	17.07%
19	161.07%	75.69%	75.63%	62.47%	12.20%	14.63%
20	170.66%	75.14%	78.18%	62.33%	7.32%	9.76%

Table 6.24: Predictive accuracy measures for the EbA model (LNTotalSkillCost)

The comparison of the two models shows that the linear regression model outperforms EbA for MMRE, MdMRE, MMER and MdMER, whereas the opposite is true for the remaining measures. On the other hand, the parametric and non-parametric tests do not provide a statistically significant difference between these measures.

6.7 Summary

In this chapter, two methods for estimating the actual effort and total cost both for core and support system projects were presented and their accuracy was evaluated. Results suggest that better estimates are obtained when cultural and leadership attributes are included in the estimation model. Specifically, the estimation of actual effort improved in 90% of the support system projects and in 50% of the core system projects, when leadership and cultural attributes were added. The bootstrap method was used to adjust and correct the estimate of the analogy cases and showed close results in the actual efforts. Total skill cost may be used as alternative evaluation for software effort estimation due to its importance and significance in predicting the cost model.

In this chapter, an evaluation of the SEEOS system was provided. The system was proved to be a useful and efficient system in a number of ways. It was compared to other techniques and models. Its architecture was also compared to others where new attributes were included and an assessment was made of their importance with respect to the role of software project development. The system was also evaluated on the basis of user feedback and IT managers were involved in suggesting ways of improving the system.

This model delivers an estimation of new projects based on the data and attributes of previous projects. Effort was analysed by regression and analogy. Regression and analogy performed better when projects were split and selected as core and support systems. As an overall analysis of the completed projects, including leadership and cultural attributes, this model appears to give better results with any methods.

CHAPTER 7

Conclusion

7.1 Introduction

This chapter summarises and concludes this thesis. It includes suggestions for future research areas on which the aims of this research have focused. It considers the general applicability of the system, that is, the areas of need to which the cost estimation and the proposed methods are useful. It also reviews the original contribution that has been made in carrying out the completed work. The chapter discusses the overall effectiveness and value of the research. Lastly, a brief review and conclusion to the thesis is presented.

7.2 Aims and Achievements

The primary aims of this research were discussed in Section 1.2 and are briefly summarised here.

1. To investigate the contribution of organisational culture and leadership factors in software project development.
2. To investigate and identify software cost estimation and determine the impact of organisation culture and leadership on the devised model.
3. To identify any software development problems relating to the needs of the Gulf States in particular and the limitations of current software project estimation methods.
4. To fully understand Case-Based Reasoning methods to improve effort estimation in software development projects.

5. To fully explore the importance of organisational culture and leadership factors in software project estimation.
6. To create a new cost estimation model that is applicable for government departments that takes culture and leadership factors into account.
7. To improve CBR and effort estimation in software development projects by capturing the organisational culture and leadership attributes using ontology-based systems.

The aim of this investigation is to develop an accurate and efficient method for software cost estimation based on Case-Based Reasoning in an effort to improve the accuracy of effort estimation by analogy. Based on this research, leadership characteristics, such as the level of interaction and relationships, decision-making, the ability to motivate team members, team members' characteristics, cultural intelligence, active thinking and communication skills are all shown to be important factors affecting software development. Consequently, when different teams are involved in a project, their cultures and backgrounds affect their effort either positively or negatively. Issues like timeliness, collaboration and team work, cultural intelligence, reward, job stability and communication all affect willingness and enthusiasm towards the success of the work in software development and in estimation software products.

These characteristics result in different problems with various degrees of severity based on the cultural impact of the leader. In the Gulf States, more than elsewhere, culture has a greater role in affecting work performance and the level of dedication found in team leaders.

From the questionnaire and interviews, it is concluded that cost estimation in the Gulf States does not conform to accepted cost models. Instead, it relies mostly on ad hoc expert judgement. Such an approach lacks data maintenance and published guidelines, resulting in variations in measurement cost among projects.

By capturing the cultural dimension, the model that is proposed is capable of accurately estimating software development costs for projects in the Gulf States. Dominant factors are not affected by the presence of various cultural backgrounds. The proposed model can be applied to other cultures and countries. Culture and leadership factors have been recognised as important in positively or negatively affecting cost estimation. The scale is applicable in showing how significant each item is for each culture.

The inclusion of leadership and culture in the cost estimation model constitutes an enhancement and refinement to CBR. It is also an improvement over models that do not take leadership and cultural backgrounds into account in their cost estimation. In this research, two models, one with cultural factors and one without cultural factors, were used. The culture-augmented model (SEEOS) results in significantly more accurate estimations than the one without the cultural attributes.

7.3 Applicability of the Model

The effort estimation model was essentially developed based on CBR which allows for the retrieval of case bases. It would thus be suitable for applications with different domain data and in different lines of business. The system is generally applicable for most IT government departments where software development projects are important to keep records of historical lessons and of knowledge. It is also applicable to other areas where the validation of particular parameters is required.

A well-defined software estimation model such as this will improve not only cost for government departments but will also enhance IT managers' knowledge and thus team functions in software development organisations. This tool will aid project managers in their work in understanding and explaining the software process and also in giving an awareness of new factors that play major roles in determining project effort and future challenges. The developed model for software cost estimation has applications in government departments in the Gulf States. Clearly, leaders are keen and enthusiastic to seriously consider more advanced models for software cost estimation. Their urgent need for proper estimation methods makes this model more suitable with fewer limitations. Thus, the attributes are shown to contribute to the accuracy of analogy.

7.4 Originality of the work

In this research, the original contribution is a novel method of effort estimation based on Case-Based Reasoning that incorporates organisational culture and leadership aspects. This method is expected to lead to improvements in the accuracy of effort estimation by analogy.

- In particular, this work uses new data based on a large number of cases obtained from organisations in the Gulf States.

- This work, with its previously stated aims, is innovative. Moreover, the difficult problem of evaluating culture and leadership, in such a way that it can be incorporated in a CBR model, has not been attempted before in such a context.
- The study shows that there was no effective model in use or existed, either in Abu Dhabi or Dubai for software cost estimation. This resulted in failures and over-runs in most of the application development projects. This research thus recommends that a software cost estimation model should be constructed.
- The development of a new model addresses the issues related to the accuracy of effort estimation. The new model includes culture, leadership and project manager's authority as attributes. These additions increase the accuracy of effort estimation.

This research is one of the first attempts to use the CBR approach for developing such a model. It builds upon and extends the work of Shepperd and Schofield (1997). It is noteworthy that no previous models focused on cultural, organisational and leadership aspects. These aspects are vitally important in this software development project (see Chapter 3). Finally, a new cost estimation model was built, based on the CBR method, and implemented in government departments and public sector organisations. It has been partially funded by the UAE University to continue the building of a national database of previous projects. The creation of the new cost estimation model takes into account the size of the project, the skills involved and the recording in the software project data of the cultural and leadership characteristics of the developed team.

Using similarity case matching retrieval, this research employs an ontology as a set of measurements and definitions of the attributes and factors. The behaviour of the model is consistent with that of other models which do not take culture and leadership into account. However, once included, these factors show a difference and result in more accurate estimations, thus enhancing CBR.

7.5 Further work

In this research a system has been developed, implemented and evaluated using collected data that depend on the classification of projects of the system. In the case of core projects, it was realised that the notion of core lacked focus, thus resulting in the grouping of diverse projects. Consequently, the collected data also lacked focus. Further study is needed to clearly identify features of core projects in order to collect data that are consistent. For applications in which the numbers of algorithms are countable such as the total number of operators and operands, and where algorithmic factors are significant, feature points would be more appropriate than function points. Many systems applications fall within these categories (Jones, 1991).

The resulting application is a stand-alone system requiring installation on the manager's computer. A way to improve this model is to make it web-enabled so that users can access it from anywhere. Currently, a change to or modification of the ontology is difficult because it is statically represented through the user interface and modification of the ontology requires the modification of the user interface. To improve the system and make the ontology dynamic means that modifications are needed at the business logic tier. For example, the use of the XML language to represent the ontology may provide the needed flexibility.

Moreover, future research should look at increasing the size of the population considered for the survey. More international organisations or more Gulf countries could participate in the survey so that a broader understanding of the measurement would be possible.

An evident improvement to this research would require that the survey be refined by providing a scale to collect more accurate data. These data should be more representative of the quality of the leader. For example, the scale should be changed from good to excellent instead of low to high. This would probably encourage the leaders to be more responsive to the survey. These current limitations of the model will form the basis of future research proposals.

7.6 Summary and Conclusions

The aims of this research were to find a suitable approach using analogy techniques to perform the retrieval of applicable cases for software development. Culture and leadership have been recognised as important in positively or negatively affecting cost estimation. The proposed model can be applied to other cultures and countries. The scales used shows how significant each item is for each culture.

The new model results show that practical benefits can be obtained from implementing this system. The model evaluation shows that IT managers at all levels are satisfied with the usefulness of this approach. To help team projects succeed, organisations need good leaders who are willing to make a difference in the team members' lives. Projects need leaders who are willing to help the team overcome their fears of being different in the workplace. The leaders need to be equipped with knowledge that allows them to find ways to help team members relate better to different cultures. Exceptional leaders are the ones who are resilient, flexible and sufficiently comfortable to be able to encourage, support and enhance the learning of team members from diverse cultures.

Culture is one of the most important aspects that affect peoples' lives, their behaviours and their thinking. Culture is not an easy concept to define and has been compared to an iceberg. "Just as an iceberg has a visible section above the waterline, and a larger, invisible section below the waterline, so culture has some aspects that are observable and others that can only be suspected, imagined, or intuited. Also like an iceberg, that part of culture that is visible (observable behaviour) is only a small part of a much bigger whole" (The Peace Corps Cross-Cultural Workbook, 2008, p. 10).

Finally, the leaders should encourage team members to work together in an atmosphere free of conflict, an atmosphere full of motivation, cooperation, teamwork and understanding. They should spread harmony, peace and tolerance among team members. Leaders need to foster the spirit of learning throughout the team members' development process.

References:

Aamodt, A. and Plaza, E. (1994) “Case-based reasoning: Foundational issues, methodological variations, and system approaches”, *AI Comm*. 7(1), pp. 39–59.

Abdel-Hamid, T. and Madnick, S. (1991) *Software Project Dynamics*, Englewood Cliffs, NJ, Prentice-Hall 1991.

Abdulla, A and Al-Homoud, A. (2001) “Exploring the implicit leadership theory in the Arabian Gulf states”, *Applied Psychology: International Review*, 50(4), pp. 506-31.

Abu-Rmaileh, S. and Hamdan, K. (2007) “Culture and its effect on student learning”. In S. Stewart and J. E. Olearski (Eds.). Proceedings of the third annual conference for Middle East teachers of science, mathematics and computing, *METSMac: UAE*, pp. 343-355.

Adair, J. (1998) *Effective leadership*: How to develop leadership skills, London: Pan.

Aha, W. (1991) “Case-Based Learning Algorithms”, Proc. *DARPA Case-Based Reasoning Workshop*.

Ahire, S. and O’Shaughnessy, K. (1998) “The role of top management commitment in quality management: an empirical analysis of the auto parts industry”, *International Journal of Quality Science*, MCB UP Ltd, 3(1), pp. 5 – 37.

Albrecht, A. (1979) “Measuring Application Development Productivity”, *PrIBM Applications Development Symp*., Monterey, California, pp. 14-17.

Albrecht, J. and Gaffney, R. (1983) “Software measurement, source lines of code, and development effort prediction: a software science validation”, *IEEE Transactions on Software Engineering*, 9 (6), pp. 639– 648.

Allen, R. and Kraft, C. (1987) *The Organizational Unconscious*, Morristown, NJ, Human Resources Institute.

Allio, J. (2006) “Strategic thinking: the ten big ideas, strategy and leadership”, 34 (4), *Emerald Group Publishing,* pp. 4-13.

Angelis, L. and Stamelos, I. (2000) “A simulation tool for efficient analogy-based cost estimation”, *Empirical Software Engineering*, 5(1), pp. 35-68.

Angelis, L., Stamelos, I. and Morisio, M. (2001) “Building a Software Cost Estimation Model Based on Categorical Data”, Proceedings of the 7th IEEE *International Software Metrics Symposium*, pp. 4-15.

Atkinson, K. and Shepperd, M. (1994) “Using Function Points to Find Cost analogies”, *European Software Cost Modeling Conference*. Ivrea, Italy.

Bailey, J. and Basili, V. (1981) “A meta-model for software development resource expenditures”, 5th International *Conference on Software Engineering*, *IEEE Press*, pp. 107-116.

Bellout, D., Harbi, H. and Hamdan, K. (2006) “Bootstrap and other Methods to Measure Learning in UAE Universities and Higher Colleges”, Proceedings of the second annual conference for Middle East teachers of science, mathematics and computing, *METSMac:* UAE, pp. 221-230.

Bergmann, R. and Schaaf, M. (2003) “Structural Case-Based Reasoning and Ontology-Based Knowledge Management: A Perfect Match?” *Journal of Universal Computer Science*, 9(7), pp. 608-626.

Boehm, B. (1981) *Software Engineering Economics*. Englewood Cliffs, N.J. Prentice-Hall, Inc.

Boehm, B., Abts, C. and Chulani, S. (2000) “Software development cost estimation approaches – A Survey”, *Science Publishers*, 10(4), pp. 177-205.

Boehm, W. B. and Sullivan, K.J. (2000) “Software economics: a roadmap”, in the future of software engineering, 22*nd* *International Conference on Software Engineering*, pp. 319-344. Viewed: 3/08<http://citeseer.ifi.unizh.ch/boehm00 software.html>.

Briand, L., El Emam, K., Surmann, D., Wieczorek, I. and Maxwell, K. (1999) “An assessment and comparison of common software cost estimation modeling techniques”, ICSE’99, *IEEE Computer Society Press*, CA, USA, pp. 313-322.

Brown, A. (1998) *Organizational Culture*, 2nd edition, London: Financial Times Prentice-Hall.

Bunin, B. (2003) *Microsoft Project 2002*, Course Technology: A division of Thomson Learning, Inc.

Charette, R. (2005) “Why Software Fails”, *IEEE Spectrum*, pp. 1-11. Viewed: 5/12/07<http://drewyates.net/cse757/whySoftwareFails.pdf>.

Claude, S. (2006) News World Communications *in Middle East Time*, Washington D.C. USA. Viewed: See http://www.metimes.com/storyview.php

Conte, S., Dunsmore, H., and Shen, Y. (1986) Software Engineering Metrics and Models. Menlo Park, California, *Benjamin Cummings*.

Cosgrave, J. (1996) “Decision-making in emergencies”, Disaster Prevention and Management, *MCB University Press* 5(4): pp. 28–35.

Crawford, S. (2007) “Successful People Are On Time!”, Ezine Articles. Viewed: 2/11/07 <http://ezinearticles.com/?id=531944>.

Cronbach, J. (1951) “Coefficient alpha and the internal structure of tests”, *Psychometrika, 16(3)*, pp. 297-334.

Delany, S., Cunningham, P. and Wilke, W. (1998) “The Limits of CBR in Software Project Estimation”, GWCBR'98, 6th German Workshop on CBR, Berlin, pp. 1-10.

Edelstein, H. (1994) “Unraveling Client Server Architectures”, *DBMS 7,* 5, 34(7).

Efron, B. (1992) “Jackknife-after-Bootstrap Standard Errors and Influence Functions”, *J.R. Statistical Society*, 54(1), pp. 83-127.

Efron, B. and Tibshirani, R. (1986) “Bootstrap Methods for Standard Errors, Confidence Intervals, and other Measures of Statistical Accuracy”, *Statistical Science*, 1(1), pp. 54-77.

Efron, B. and Tibshirani, J. (1993) *An Introduction to the Bootstrap*, New York, Chapman and Hall.

Elfenbein, A. and O’Reilly, C. (2002) “Fitting the effects of relational demography and person-organization fit on group process and performance”, Research Paper No. 1728, Harvard Business School.

Fairholm, G. (1991) *Values of Leadership*: Toward a new philosophy of leadership, Greenwood, NY, pp. 4-150.

Farquhar, A., (1970) “A Preliminary Inquiry into the Software Estimation Process”, *Rand Corporation*, Santa Monica, California.

Fenton, E. and Pfleeger, L. (1997) *Software metrics:* a rigorous and practical approach, PWS Publishing, Boston, MA.

Forrester, J. W., (1968) *Principles of systems*, 2nd edition. Cambridge, MA: Productivity Press.

Freiman, R. and Park, D. (1979) “PRICE software model version 3: An overview”, Proceedings of IEEE-PINY workshop on *Quantitative Software Models*, *IEEE Cat*., p. 3241.

Futrell, R., Shafer, D. and Shafer, L. (2002) *Quality Software Project Management*, 3rd edition, Prentice-Hall, New Jersey: Achieving Quality Software.

Gabel, R., Dolan, S. and Cerdin, J. (2005) “Emotional intelligence as predictor of cultural adjustment for success in global assignments”, Career Development International, 10 (5), *Emerald Group Publishing Limited*, pp. 375-395.

Garmus, D. (2006) “The Principles of Sizing and Estimating Projects Using International Function Point User Group (IFPUG) Function Points”, *David Consulting Group*, Viewed: 1/7/07 <http://www.davidconsultinggroup.com>.

Gardner, W (1990) “On Leadership”, *The Free Press*, New York, Viewed: 1/7/07 <http://develop-net.com/articles/sept1997.html>.

Gordon, J. and Helmer, O. (1964) "Report on a Long-Range Forecasting Study", *Rand Corporation*, Santa Monica, California, pp. 1- 71.

Gray, J. (1981) "The Transaction Concept: virtues and limitations", Tandem Computers Incorporated, 7th International Conference*, IEEE Press*, Technical Report 81.3

Gruber, T. (1993) "Toward Principles for the Design of Ontologies Used for Knowledge Sharing", Technical Report. KSL 93-04, *In International Journal Human-Computer Studies* 43, pp. 907-928.

Hamdan, K., Abu Sitta, F., Moses, J. and Smith, P. (2005) "An Investigation into the Gulf States Government Approaches to Software Development and Effort Estimation", *BCS-SQM*, UK

Hamdan, K., El Khatib, H., Moses, J. and Smith, P. (2006) "A Software Cost Ontology System for Assisting Estimation of Software Project Effort for Use with Case-Based Reasoning", Innovations in Information Technology, *IEEE.*

Hartenian, L. (2003) "Team member acquisition of team knowledge, skill and abilities", Performance Management: *International Journal*, 9(1/2), pp. 23-30.

Heemstra, J. (1992) "Software cost estimation", *Information and Software Technology*, 34(10), pp. 627-63.

Harold, B. (2006) "Developing Leaders: A Middle Eastern Perspective", Viewed: 1/11/2007 < http://www.aare.edu.au/06pap/har06550.pdf>.

Hills, H. (2004) "Collaborative Decision-Making: how to make it work", *Training Journal,* UK, pp. 18-22.

Hofstede, G. (1991) *Culture and Organizations*: Software of the Mind. London: McGraw-Hill.

Hofstede, G. and Hofstede J., (2005) *Cultures and Organizations*: Software of the mind revised and expanded, 2nd Edition. New York: McGraw-Hill.

Hughes, T. (1996) "Expert Judgement as an estimating method", *Information and Software Tech*nology, 38(2), pp. 67-75.

Hunt, J. and Larson, L. (1975) *Leadership Frontiers*: A Role-Making Model of Leadership in Formal Organizations, Kent State University Press, Ohio.

ISBSG, (2003) "International Software Benchmarking Standards Group, Data Repository", Release 8, Viewed: 7/05 <http://www.isbg.org.au>.

Janulaitis, V. (2005) "Information Technology Comparative Compensation Study: Salary Survey", Janco Associates, Inc., Park City, UT, pp. 5-33.

Jensen, R. W. (1984) “A comparison of the Jensen and COCOMO and estimation models”, *Proceedings of the International Society of Parametric Analysts*, pp. 96-106.

Johnson, J. (2006) “Why project failures?”, Viewed: 8/07 <http://www.infoq.com/articles/chaos-1998-failure-stats>.
Jones, C. (1998), *Estimating Software Costs*. New York: McGraw-Hill Education.

Jones, C. (1991) *Applied Software Measurement*: Assuring Productivity and Quality. New York: McGraw-Hill.

Jorgensen, M. and Shepperd, M. (2007) “A Systematic Review of Software Development Cost Estimation Studies”, *IEEE Transactions on Software Engineering*, 33, pp. 33-53.

Kadoda, G., Cartwright, M., Chen, L. and Shepperd, M. (2000) “Experiences using Case Based Reasoning to predict software project effort”, Proceedings of the 4th *International Conference*, pp. 17-19.

Kadoda, G., Cartwright, M. and Shepperd, M.J., (2001) “Issues on the effective use of CBR technology for software project prediction”, Proceedings of the 4th International Conference on CBR*, CCBR*, Canada, pp. 276-290.

Kaplan, A., Skogstad, L. and Girshick, A. (1949) “The Prediction of Social and Technological Events”, Public Opinion Quarterly, 14(1), *Rand Corporation*, pp. 93-110.

Kaner, C. and Bond, W. (2004) “Software Engineering Metrics: What Do They Measure and How Do We Know?”, 10th International Software Metrics, Symposium, pp. 1-12.

Kemerer, F. (1987) “An Empirical Validation of Software Cost Estimation Models”, Research Contributions, *ACM*, 30(5), pp. 416-429.

Kemerer, F. (1997) *Software Cost Estimation Models*, Software Project Management Readings and Cases. Boston: McGraw-Hill.

Kitchenham, B., Fenton, N. and Pfleeger, S. (1995) “Towards a framework for software measurement validation”, *IEEE Transaction Software Engineering*, 21(12), pp. 929-944.

Kitchenham, B., Hughes, T. and Linkman, S. (2001) “Modeling Software Measurement Data”, *IEEE Transaction Software Engineering*, 27(9), pp. 788-804.

Kitchenham, B., Pfleeger, S., McColl, B. and Eagan, S. (2002) “An Empirical Study of Maintenance and Development Estimation Accuracy”, *Journal of Systems and Software*, 64, pp. 57-77.

Kitchenham, B. and Taylor, R. (1984) “Software Cost Models”, *ICL Technical Journal*, pp. 73–102.

Kitchenham, B., Travassos, G., von Mayrhauser, A., Niessink, F., Schneidewind, N., Singer, J., Takada, S., Vehvilainen, R., and Yang, H. (1999) “Towards an ontology of software maintenance”, *Journal of Software Maintenance*, New York, NY, 11(6), pp. 365– 389.

Kohavi, R. (1995) “A study of cross-validation and bootstrap for accuracy estimation and model selection”, Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence 2 (12): *San Mateo*, pp. 1137–1143.

Kolodner, J. (1992) “An introduction to case-based reasoning”, *Artificial Intelligence Review*, 6(1), pp. 3-34.

Kolodner, J. (1993) *Case-Based Reasoning*: Morgan-Kaufmann Publishers, Inc. California, USA.

Krishnan, M. (1998) “The role of team factors in software cost and quality”: An empirical analysis, Information Technology & People, *MCB University Press*, 11(1), pp. 20-35.

Leake, D. (1996) *Case-based reasoning*: experiences, lessons, and future directions, AAAI Press, Menlo Park, California, USA.

Lederer, A. and Prasad, J. (1998) “A Causal Model for Software Cost Estimating Error”, *IEEE Trans. Software Engineering*, 24(2), pp. 137-148.

Lederer, A. and Prasad, J. (1992) “Nine Management Guidelines for Better Cost Estimating”, *Communications, ACM*, 35(2): pp. 51-59.

Leung, H., and Fan, Z. (2003) “Software Cost Estimation”, Hong Kong Polytechnic University. Viewed: 3/06 <ftp://cs.pitt.edu/chang/handbook/42b.pdf>.

Longstreet, D. (2005) “Function Points Analysis Training Course”, Longstreet Consulting Inc. Viewed: 12/06 < http://www.softwaremetrics.com /training.htm>.

Madachy, R. (1994) “A software project dynamics model for process cost, schedule and risk assessment ”, *University of Southern California*, p. 127.

Madar Research Group (2003), “Dubai Knowledge Economy 2003-2008”, *Madar Research Group*, Dubai, Madar Research Special Report Volume I. Viewed: 5/6/05 <http://MadarResearch.com>.

Madar Research Group (2006), “Dubai Knowledge Economy 2003-2008”, *Madar Research Group*, Dubai, Madar Research Special Report Volume II. Viewed: 1/1/07 <http://MadarResearch.com>.

MacDonell, S. and Shepperd, M. (2003) “Combining Techniques to Optimize Effort Predictions in Software Project Management”, *Journal of Systems and Software*, 66, pp. 91-98.

Mair, C., Kadoda, G., Lefley, M., Phalp, K., Schofield, C., Shepperd, M., and Webster, S. (2000) “An investigation of machine learning based prediction systems”, *Journal of Systems Software*, 53(1), pp. 23-29.

Masticola, P. (2007) “A Simple Estimate of the Cost of Software Project Failures and the Break-even Effectiveness of Project Risk Management”, Siemens Corporate Research, *Computer Society*, 29th International Conference on Software Engineering Workshops, IEEE.

Moses, J. (1997) “Cohesion Prediction Using Information Flow”, *University of Sunderland*, Thesis, UK.

Moses, J., Farrow, M. and Smith, P. (2002) “Statistical Methods for Predicting and Improving Cohesion Using Information Flow”, An Empirical Study, *Software Quality Journal* 10(1), pp.11-46.

Mukhopadhyay, T., Vicinanza, S. and Prietula, J. (1992) “Examining the feasibility of a case-based reasoning model for software effort estimation”, *MIS Quarterly*, 6(2), pp.155-171.

Myrtveit, I., Stensrud, S. and Shepperd, M. (2005) “Reliability and Validity in Comparative Studies of Software Prediction Models”, *IEEE Transition on Software Engineering*, 31(5), pp. 380-391.

NASA JSC (2002) “Parametric Cost Estimating Handbook”, *Johnson Space Centre*. Viewed: 12/05 <http://cost.jsc.nasa.gov/PCEHHTML/pceh.htm>.

Norden, V. (1958) “Fitting for a Model of Applied Research and Development Scheduling”, *IBM Journal of Research and Development,* 2, pp. 232-48.

Neal, M., Finlay, J. and Tansey, R. (2005) “My father knows the minister”: A comparative study of Arab women’s attitudes towards leadership authority, 20(7), pp. 478-497.

Nelson, A. (1966) *Management Handbook for the Estimation of Computer Programming Costs*, Systems Development Corp, Santa Monica, CA.

Osborn, N. (2006) “CQ-Cultural Intelligence: Another Aspect of Emotional Intelligence”. Viewed: 7/07 <www.team-inc.net/Resources/Docs/ Cultural% 20Intelligence.pdf>.

Osteraker, C. (1999) “Measuring motivation in a learning organization”, *Journal of Workplace Learning*, 11(2), pp. 73–77.

Pallant, J. (2005) *SPSS survival manual*, 2nd edition, Open University Press, McGraw-Hill House, UK.

Park, R. (1988) “The Central Questions of the PRICE Software Cost Model,” 4th COCOMO Users’ Group Meeting.

Park, R. (1992), "Software Size Measurement: A Framework for Counting Source Statements", *CMU/SEI-92-TR-20*, Technical Report. Viewed: 6/5/08 <http://www.sei.cmu.edu/pub/documents/92.reports/pdf/tr20.92.pdf>.

Patterson, D., Galushka M. and Rooney, N. (2005) "Characterization of a novel indexing technique for case-based reasoning", *Artificial Intelligence Review*, 23(4), pp. 359-393.

Plekhanova, V. (1999) "Applications of the Profile Theory to Software Engineering and Knowledge Engineering", Proceedings of the 12th *International Conference on Software Engineering and Knowledge Engineering*, pp. 33-141

Plekhanova, V. (2000) "On the Compatibility of Contemporary Project Management Tools with Software Project Management", proceedings of the 4th world multi-conference on systemics, *Cybernetics and Informatics*, pp.1-6.

Poole, B. (2006) "Values and Culture". Viewed: 1/12/07 <http://www.muw.edu/ nursing /tupelo/Culture.html>.

Pressman, R. (1997) Software Engineering, A Practitioner's Approach, 4th edition, McGraw-Hill series in computer science.

Putnam, L. (1978) "A general empirical solution to the macro software sizing and estimating problem", IEEE Trans. on Software Engineering, 4(4), pp. 345-61.

Putnam, L. and Myers, W. (1992) *Measures for Excellence*, Yourdon Press Computing Series.

Rahim, M. and Minors, P. (2003) "Effects of emotional intelligence on concern for quality and problem-solving", *Emerald Research*, 18(2), pp. 150-155.

Reifer, D., Boehm, B., and Chulani, S. (1999) "The Rosetta Stone: Making COCOMO 81 Files Work With COCOMO II", D, Crosstalk, *The Journal of Defence Engineering*, pp. 1-10.

Richardson, G. (1991) *System Dynamics: Simulation Modeling and Analysis*, Fishwich and Luker, eds., Springer-Verlag.

Ringer, T. (2007) "Viewpoint leadership for collective thinking in the workplace", *Team Performance Management*, Emerald Group Publishing Limited, 13(3/4), pp. 130-144.

Rubin, H. (1983) "Macro-estimation of software development parameters: The ESTIMACS System", N.Y, IEEE, pp. 109 -118.

Ryder-Smith, J. (1999) "Talking their language", Jocelyn Dearden Management, *Team Performance Management*, 5(2), Bristol, UK, pp. 78-81.

Sadri, G. (1996) "Reflections: The impact of downsizing on survivors, some findings and recommendations", *Journal of Managerial Psychology*, Fullerton, USA, 11(4), pp. 56-59.

Sarros, C., Cooper, K. and Hartican, M. (2006) “Leadership and Character”, Leadership and Organization, *Development Journal*, Emerald Group Publishing Limited, 27, pp. 682-699.

Schein, E. (2004) *Organizational Culture and Leadership*, 3rd edition, Joddry-Bass, John Wiley & Sons, pp. 129-225.

Schneider, C. and Barsoux, J. (2003) *Managing across cultures*, Pearson Education, 2nd Edition, England.

Sentas, P., Angelis, L., Stamelos, I., and Bleris, G. (2005) “Software productivity and effort prediction with ordinal regression”, *Information and Software Technology* 47, pp. 17–29.

Seymour-Smith, C. (1992) *Dictionary of Anthropology*, G.K. Hall and Company Boston, USA, Macmillan Press Ltd., London and Basingstoke.

Shepperd, M. (2003) “Case-Based Reasoning and Software Engineering”, *Managing Software Engineering Knowledge*, in Aurum et al. (ed), Springer-Verlag.

Shepperd, M. and Schofield, C. (1997) “Estimating Software Project Effort Using Analogies”, *IEEE Transactions on Software Engineering*, 23(12), pp. 736-743.

Shepperd, M., Schofield, C. and Kitchenham, B. (1996) “Effort Estimation Using Analogy”, 18th *International Conference on Software Engineering*, IEEE, Berlin, pp. 170-178.

Sheth, A. and Lytras, M. (2007) *Semantic Web-Based Information Systems*: State of the Art Applications, Cybertech Publishing, London, UK.

Snell, D. (1997) “Software Cost Estimation”, Bournemouth University, Viewed: 1/12/07 <http://www.ecfc.u-net.com/cost/index.htm>.

Solomon, N., Boud, D., Leontios, M. and Staron, M. (2001) “Researchers are learners too: collaboration in research on workplace learning”, *Journal of Workplace Learning*, MCB University Press, 13(7/8), pp. 274-281.

Sommerville, I. (1995) Software Engineering, 5th Edition, Harlow, Addison-Wesley, UK.

Stamelos, I. and Angelis, L. (2001) “Managing uncertainty in project portfolio estimation”, *Information and Software Technology*, Elsevier Science, 43, pp. 759-768.

The Standish Group (1994) “CHAOS-Research Report”. Viewed: 1/12/07 <http://standishgroup.com / sample_research/ PDFpages/extreme_chaos.pdf>.
Stellman, A. and Greene, J. (2006) *Applied Software Project Management*, O’Reilly Media Inc. CA.

Stewart, J. (1989) “Worker Reward Schemes”, Management Decision, 27(4), MCB UP Ltd, pp. 1-25.

Swigger, K., Alpaslan, F., Brazile, R. and Monticino, M. (2003) “Effects of culture on computer-supported international collaborations”, *Elsevier Ltd International Journal of Human Computer Studies*, 60, pp. 365–380.

Tausworthe, R. (1981) “Deep Space Network Software Cost Estimation Model”, *Jet Propulsion Laboratory Publication*, pp. 81-87.

The Peace Corps Cross-Cultural matters Workbook (2008)”, Viewed: 15/2/08 <http://www.peacecorps.gov/library/pdf/T0087_culturematters.pdf>.

UNStats (2005) “Global E-government Department of Economic and Social Affairs”, Division for Public Administration and Development Management Readiness Report, NY. Viewed: 6/12/06 <http://unpan1.un.org/intradoc/groups/public/ documents/un/unpan021888.pdf>.

Walston, C. and Felix, P. (1977) “A method of programming measurement and estimation”, *IBM Systems Journal*, 16(1), pp. 54-73.

Watson, G., and Gallagher, K. (2005), *Managing for Results*, Chartered Institute of Personnel and Development (CIPD), 2nd Edition.

Yousef, D. (1998) “Predictors of decision-making styles in a non-western country”, *Leadership and Organization Development Journal*, 19(1), pp. 366-373.

Appendix A

Pilot Survey

A survey of software development effort estimation in private and governmental agencies in the United Arab Emirates

Currently I am conducting a study to investigate the process used to estimate the effort of developing and maintaining Information Systems/Software in government and semi-government departments in the Gulf region. The objectives of this study are to determine the current state of the art in software effort estimation techniques to help identify methods and research directions for improving software effort estimation and control.

I wish to look at Information Systems/software development in the UAE. My approach is to identify a set of development and maintenance projects, which have been completed (or are close to completion), to determine when and how the effort costs were estimated, and then to look at how the estimated effort relates to the actual effort.

I am interested in all systems that contain software. I will be looking at governmental and commercial systems, maintenance and development projects, embedded and information systems.

This document contains a set of topics and questions that are intended to gather information about a specific project, which has been undertaken by your organisation.

The following questions will help me to determine the effort spent on information systems and software development. You can be assured that all replies will be treated in the strictest confidence, and that no organisation or individual will be named, without prior approval, in any reports produced as a result of this questionnaire. If you are interested in receiving a copy of the collated results from this study, please let me know during the interview.

Please select the most important or the largest recent project you had been involved with.

To make further inquires please feel free to contact me at the following Address:

Khaled Hamdan
P O Box 17172
United Arab Emirates University, IT Unit-UGRU
Al Ain
United Arab Emirates
Khamdan@uaeu.ac.ae
Tel: 971(50)6436350

1. Was the project developed totally in-house or was it contracted to an external vendor?

   ☐ In-house
   ☐ Out sourced

2. Does your department have developed or used any of the following systems?

		Yes	No
a)	Enterprise Recourse Planning System	☐	☐
b)	Business related Systems	☐	☐
c)	Control System	☐	☐
d)	Decision Support System	☐	☐
e)	Other (Please state) ___________________________		

3. What was expected to be delivered at the completion of the project?

   ☐ A new system was required
   ☐ A major upgrade: a currently existing system was upgraded, requiring extensive amounts of
   new software to be developed
   ☐ A minor upgrade: a currently existing system was upgraded required (some) software
   ☐ A system was updated by modify existing software

4. Which one of the following methods have you selected while allocating the budget for the IT projects?

   ☐ Estimated the software development effort separately and add it to the total cost of the project
   ☐ Didn't estimate the software development effort and instead looked at the project as a whole

5. What was the contracted effort?

   ☐ Time based contract
   ☐ Based on Deliverables

6. How close was the contracted effort to the bids? (Please state).

   ☐ More
   ☐ Less
   ☐ Acceptable (variance between actual effort and estimated one are negligible)
   ☐ Not applicable (N/A)

7. In your opinion, what were the reasons for the discrepancy between the actual effort and the estimates?

- ☐ Inability to anticipate skill of project team members
- ☐ Overlooked tasks
- ☐ Lack of an adequate methodology or guidelines for estimating
- ☐ Lack of historical data regarding past estimates and actual performance
- ☐ Lack of project control comparing estimates and actual performance
- ☐ Inability to tell where past estimates failed
- ☐ Frequent requests for changes by users
- ☐ Performance reviews don't consider whether estimates were met
- ☐ Missing Data
- ☐ Other (please state) ________________________

8. At what stage of the project life cycle did this effort estimation occur?

- ☐ Beginning of each phase
- ☐ Initial project plan
- ☐ Prior to project award (before approval)
- ☐ Prior to implementation stage
- ☐ After system requirement analysis phase
- ☐ End of stage 1 out of 4 stages

9.1 Were any re-estimates of effort performed during development?

☐ Yes ☐ No

9.2 If Yes to 9.1, then please identify when the re-estimating procedures were invoked (see list below). If possible, indicate when (in the development life cycle) the re-estimates occurred, and the result of the re-estimates. (Please tick appropriate box)

- ☐ Informal re-estimates performed during development.
- ☐ Formal re-estimates performed at pre-defined milestones.
- ☐ An amendment changed the system being built and a re-estimate was required.
- ☐ Other (Please state)__________________________________

9.3 Which items were included in the re- estimates? (Check all that apply)

- ☐ Software
- ☐ Hardware
- ☐ Number of delivered units
- ☐ Testing
- ☐ Integration
- ☐ Documentation
- ☐ Training
- ☐ Other (please state) _________________________

10. List the job titles of the people whom involved in the estimation process.

11. Did the people involved in the estimation process have any previous experience in software effort estimation before they were called upon to estimate for this project?

☐ Yes ☐ No

12. Was any training in project management provided for individuals?

☐ Yes ☐ No

13. Is such training required in your opinion?

☐ Yes ☐ No

14. If training was provided, do you think the training improved your organisation's ability to arrive at accurate effort estimates?

☐ Yes ☐ No

15. I would be grateful for any other information that you could offer, in order to clarify your responses, or to make further suggestions for the study. This may include questions that were not asked or were inappropriate to your department.

16. Would you be willing to take part in a further interview?

☐ Yes ☐ No

17. Would you like a copy of the results of the survey?

☐ Yes ☐ No

Thank you for taking part in this survey and for your co-operation.

APPENDIX B

Survey 1

UNITED ARAB EMIRATES UNIVERSITY
Information Technology Program (UGRU)
جامعة الإمارات العربية المتحدة
برنامج تقنية المعلومات

To Whom It May Concern

Re: Interview /Survey with IT Project Manager

Dear Sir or Madam:

Mr. Khaled Hamdan is an outstanding lecturer at the United Arab Emirates University – UGRU IT Unit. He is well liked by his students and his colleagues. I hope that you will be able to assist him in obtaining information which he is investigating the process used to estimate the effort of developing and maintaining systems involving significant amount of software.

I do appreciate that your time is precious, and as such, taking part in a survey as part of his research may not rank high on your daily priority list. However, I would be thankful if he may meet with you to discuss your organization's software effort estimation techniques.

You can be assured that all the collected data will be treated in the strictest confidence, and that no organization or individual will be named, without prior written approval, in any reports produced as a result of his research and survey.

It is hoped that the results from the survey will help to structure the development of support tools, which can assist in the selection of methods and techniques used in large-scale organizations. The study will provide useful information to organizations such that to enable improvements in effort estimation and I am happy to provide you with a copy of the collated results from this study.

I appreciate your cooperation and look forward for Mr. Khaled Hamdan to have an opportunity to meet you in the near future.

Best Regards,

Dr. Mohsen Raessi
Program Head - Information Technology

تلفون: 3-7042-310/7656-030 (971) فاكس: 3-7658-443 (971) ص.ب.: 17172 العين – الإمارات العربية المتحدة
Telephone: (971) 3-7042-310/7656-030 Fax: (971) 3-7658-443 P.O. Box: 17172 Al-Ain, UAE

1. *These questions aim to identify the type of project your organisation was involved in.*

1.1 Does your department have responsibility for the following types of system (Please see below for more description)

	Yes	No
a) Enterprise Resource Planning (ERP)	☐	☐
b) Document Management Systems	☐	☐
c) Help Desk System	☐	☐
d) Decision Support System (Data Warehouse or Data-mart)	☐	☐
e) Intranet Applications if yes please specify________________	☐	☐
f) eServices (automated services that are provided by the government department to its clients if yes please specify________________	☐	☐
g) Other (Please state) ________________		

1.2 What was expected to be delivered at the completion of the project?

- ☐ Fully functional software developed from scratch
- ☐ Readymade software with major customization
- ☐ A major upgrade: a currently existing system was upgraded, requires extensive amounts of customization.
- ☐ A minor upgrade: a currently existing system was upgraded requires few customization

1.3 Which of the following items your organisation considers when determining or measure the size of your projects?

- ☐ Person Months (man-days)
- ☐ Budget
- ☐ Functionality
- ☐ Lines of code (LOC)
- ☐ Function Points
- ☐ Feature Points
- ☐ Number of bubbles on a data flow diagram (DFD)
- ☐ Number of entities on entity relationship diagram (ERD)
- ☐ Count of process / control boxes on a structure chart
- ☐ Amount of documentation
- ☐ Number of objects, attributes and services on an object diagram
- ☐ Other ________________

1.4 Based on the above checked (marked) items, please state below how do you consider the project Small, Medium, Large and very Large

Person Months (man-days):

Experience: ☐ Junior ☐ senior ☐ Principal ☐ Distinguished

Duration ____

Budget :

Small <= ____ $, ____$ < Medium <= ____$, ___$ < Large <= ____$, ____$ < Very Large <= ____$

Functionality:

Small <= ____ # , ____# < Medium <= ____# , ____# < Large <= ____# , ____# < Very Large <= ____#

 # indicates to the number of functionality in a project - Simple - Intermediate - Complex - V. Complex

Lines of Code (LOC):

Small <= __ LOC , __LOC < Medium <= __LOC , __LOC < Large <= ___LOC , ___LOC < V. Large <= ___LOC

Function Points (System Requirements):

Small <= ____ # , ____# < Medium <= ____# , ____# < Large <= ____# , ____# < Very Large <= ____#

Feature Points (User Requirements):

Small <= ____ # , ____# < Medium <= ____# , ____# < Large <= ____# , ____# < Very Large <= ____#

Number of bubbles(details) on a data flow diagram (DFD / UML):

Small <= ____ # , ____# < Medium <= ____# , ____# < Large <= ____# , ____# < Very Large <= ____#

Number of entities on entity relationship diagram (ERD):

Small <= ____ # , ____# < Medium <= ____# , ____# < Large <= ____# , ____# < Very Large <= ____#

Count of process / control boxes on a structure chart:

Small <= ____ # , ____# < Medium <= ____# , ____# < Large <= ____# , ____# < Very Large <= ____#

Amount of documentation:

Small <= ____ # , ____# < Medium <= ____# , ____# < Large <= ____# , ____# < Very Large <= ____#

Number of objects (classes), attributes and services(methods) on an object diagram:

Small <= ____ # , ____# < Medium <= ____# , ____# < Large <= ____# , ____# < Very Large <= ____#

Other:

Small <= ____ # , ____# < Medium <= ____# , ____# < Large <= ____# , ____# < Very Large <= ____#

2. These questions aim to gather information about projects that were primarily contracted to external vendors. Please fill-in this section for each project separately that you have been managing and is the most important/recent project you have been involved.

2.1 Which project do you want to talk about? Please indicate____.

2.2 Was the project developed totally in-house or was it contracted to an external vendor?

- ☐ Within your organisation (in-house)
- ☐ Contracted out (out sourced)

2.3 What was the contracted scope?

- ☐ Consultancy Services
- ☐ Software development and Implementation
- ☐ Hardware purchase and Implementation
- ☐ Other (please state) ________________________

2.4 How close were the estimated efforts to the bids contracts? ● Please check only one answer

- ○ The estimated efforts were greater than 100% of the bids
- ○ The estimated efforts were greater than 50% of the bids
- ○ The estimated efforts were greater than 25% of the bids
- ○ The estimated efforts were greater than 15% of the bids
- ○ The estimated efforts were greater than 10% of the bids
- ○ The estimated efforts were greater than 5% of the bids
- ○ The estimated efforts were same or close as the bids
- ○ The estimated efforts were less than 5% of the bids
- ○ The estimated efforts were less than 10% of the bids
- ○ The estimated efforts were less than 15% of the bids
- ○ The estimated efforts were less than 25% of the bids
- ○ The estimated efforts were less than 50% of the bids
- ○ The estimated efforts were less than 100% of the bids

2.5 What were the reasons for any discrepancy between the estimated efforts and the bids proposals? Rank your answers with 1st equals to highest cause for the discrepancy and 7th the least/lowest reason of the discrepancy

- ☐ Initial requirements were vague
- ☐ New requirements have been identified in later stages
- ☐ Contractor underestimated work required
- ☐ Contractor suggested additional enhancements
- ☐ Overlooked tasks (offered more than expected)
- ☐ Lack of an adequate methodology or guidelines for estimating
- ☐ Lack of historical data regarding past estimates and actual performance
- ☐ Other (Please state). ____________________________

2.6 How accurate were the cost estimates compared to the actual project cost? Rank your answers with 1st equals to likely cause for the discrepancy and 8th the least/lowest reason of the discrepancy

- ☐ Overlooked tasks
- ☐ Lack of an adequate methodology or guidelines for estimating
- ☐ Lack of setting and review of standard durations for use in estimating
- ☐ Lack of historical data regarding past estimates and actual performance
- ☐ Lack of project control comparing estimates and actual performance
- ☐ Frequent requests for changes by users
- ☐ Performance reviews don't consider whether estimates were met
- ☐ Poor Requirements- Lack of cooperation & Experience of Vendor
- ☐ Other (please state). ________________________

3. These questions are intended to determine what types of effort estimates were performed
<u>Please fill-in this section for each project separately that you have been managing for the last five years, if any</u>

3.1 During the course of the project, identify when formal effort estimates occurred. For each estimate, identify:

a. At what stage of the project life cycle did this effort estimation occur?
- ☐ Prior to project award (before management approval)
- ☐ After the Bids review and approvals
- ☐ Other (Please state).____________________________________

b. Which parameters were included in the estimation process?
- ☐ Software/systems delivered functionality
- ☐ Project Duration/man days
- ☐ Other (Please state).____________________________________

c. For what purpose was the effort estimate being done?
- ☐ Budgetary approval
- ☐ Internal manpower requirement
- ☐ Gain client requirement
- ☐ Planning and project resource / allocation
- ☐ Other (please state). ________________________

3.2 Were any re-estimates of effort performed during development?

☐ Yes ☐ No

3.3 If Yes to 3.2, then please identify when the re-estimating procedures were invoked (see list below). If possible, indicate when (in the development life cycle) the re-estimates occurred, and the result of the re-estimates. (Please tick appropriate box)

- ☐ Informal re-estimates performed during development.
- ☐ Formal re-estimates performed at pre-defined milestones.
- ☐ An amendment changed the system being built and a re-estimate was required.
- ☐ Variation order
- ☐ Re-estimation were not performed
- ☐ Other (Please state).____________________________________

3.4 Which items were included in the re- estimates? (Check all that apply)

- ☐ Software
- ☐ Hardware
- ☐ Number of delivered units (Scope of work)
- ☐ Testing
- ☐ Integration
- ☐ Documentation
- ☐ Training
- ☐ Other (please state) _________________________

4. These questions are to determine who was involved in the effort estimation, and the roles, which they played. <u>Please fill-in this section for each project separately.</u>

4.1 Please tick the box that best describes your responsibilities

- ☐ Technical
- ☐ Planning & Quality Analyst
- ☐ Database Manager
- ☐ IT Consultant
- ☐ IT Engineer
- ☐ IT Manager
- ☐ System Analyst
- ☐ Finance Manager
- ☐ General Manager
- ☐ Strategic Planning
- ☐ Other ________________________

4.2 Identify the numbers of people involved in the effort estimation process and give their positions in the organisation?

- ☐ How many people involved in effort estimation ________________
- ☐ What are their positions in the organisation ______________ ____________ _____________ ____________

4.3 How were the individuals involved in the effort estimation included as part of the process?

- ☐ Informal consultation (e.g., person in charge of estimation walked into a manager's office to solicit an opinion).
- ☐ Formal consultation with project staff (e.g., a meeting was organised for purpose of performing the effort estimate).
- ☐ Sign off authority (Memorandum of understanding)
- ☐ Other (Please state) ________________________________

4.4 Did the people involved in the estimation process have any previous experience in software effort estimation before they were called upon to estimate for this project? If yes, was their experience useful in arriving at accurate estimates?

☐ Yes ☐ No

4.5 Is such training required in your opinion?

☐ Yes ☐ No

4.6 If training was provided, do you think the training improved your organisation's ability to arrive at accurate effort estimates?

☐ Yes ☐ No

4.7 Did the training include effort estimation?

☐ Yes ☐ No

I would be grateful for any other information that you could offer, in order to clarify your responses, or to make further suggestions for the study. This may include questions that were not asked or were inappropriate to your department.

____

____

____

____

____

____

____

____

5. Would you be willing to take part in a further interview?

☐ Yes ☐ No

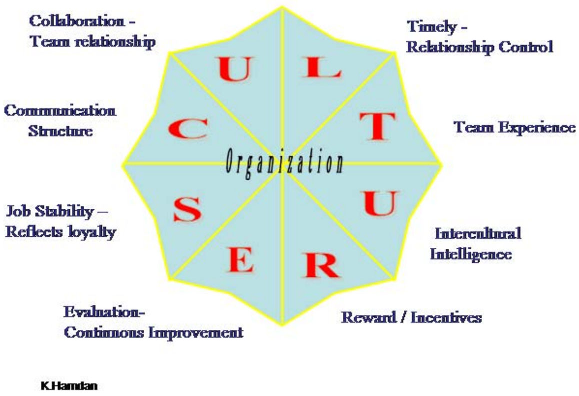
6. Would you like a copy of the results of the survey?

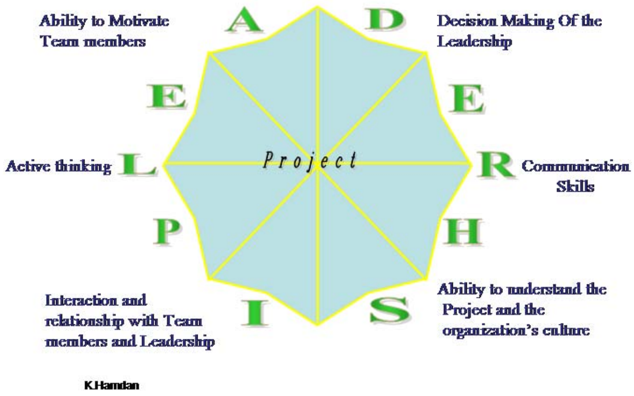
☐ Yes ☐ No

Thank you for taking part in this survey and for your co-operation.

APPENDIX C

Survey 2





A survey of software development effort estimation in private and governmental agencies in the United Arab Emirates

The purpose of this questionnaire is to invite you to join this research project. We are researching issues for project management for my Dissertation. Please email me your comments at: khamdan@uaeu.ac.ae

Your valuable assistance and knowledge will provide significant data that are necessary to fulfil the research aims of this study. Because your expertise in this field, I would be grateful in receiving a positive response to this request.

Thank you for your time and involvement in this important research project.

I. Organisation Related Data

A. Organisation Name __________________________ Region: __________(UAE)

B. Organisation Size (Number of Employees) _______

C. Organisation Type (Check what is applicable)

- ☐ Public (non-Profit)
- ☐ Private (Profit)
- ☐ Semi-government
- ☐ Other ____

D. Organisation's Line of Business (Check what is applicable)

- ☐ Medical
- ☐ Governmental Services
- ☐ Communication
- ☐ Public Services
- ☐ Tourism Services
- ☐ Education
- ☐ Oil & Gas or Energy
- ☐ Others _____________________________________

E. Organisation Type?

- ☐ Project Oriented (Project Manager has the highest power in making decisions)
- ☐ Matrix (Project Manager has moderate power in making decisions)
- ☐ Functional (Project Manager has lowest level of power in making decisions)

Please arrange for filling this survey by the following individual for each project:

☐ Executive (Functional) Manager ☐ Project Manager ☐ Team Member

II. These questions aim to identify the type of project your organisation was involved in.

A. Project Name _______________

B. Project (Application) Type (Check what is applicable)
☐ Core (**Unique features, to help achieve mission and vision of the organisations)**
☐ Supporting (**Systems which support the internal (shared) services in the organisation)**

C. The Estimated Project development duration?

_______ Months Start Date: _____________ Finish Date: _____________

D. The Actual Project development duration?

_______ Months Start Date: _____________ Finish Date: _____________

E. The Estimated and Actual project cost? Estimated Cost: __________ Actual Cost: _____________

F. The estimated and actual Effort (mandays)? Estimated Effort: ________ Actual Effort: __________

G. Number of Business Users (backend users) :

H. Number of Clients (if different from the business users):

I. Average Number of Transactions:

J. Application Specific Information:

   a. Line of Code (LOC) : _____
   b. Process: ________
   c. Objects / Classes Diagrams: _____
   d. Tables/Entities: ______
   e. Number of bubbles (DFD / UML) ________
   f. Reports / Forms: __________________
   g. Function Points (FPs): _____________

K. Technology has been used

   a. Development Tools (Programming Language):______
   b. (RDBMS): ____________
   c. Operating systems used: ___________

L. Application Architecture:-

☐ Stand alone ☐ Two Tier ☐ Three Tier ☐ Four Tiers

Please specify each Tier ___________________________________

M. The skills set and accumulator year of experience for each skill that?

Skill Type	Accumulated Experience *Please specify the # for each category*			Cost *Please specify the cost for each category*		
	Junior <5	Senior 5-10	Principal > 10	Junior <5	Senior 5-10	Principal > 10
Application Manager						
Business Advisor						
Technical Advisor/ System Analyst /Software Engineering						
Programmers/web developer						
Database/system/infrastructure Adm.						
Project Manager						
Other ________________________						

Application Manager

Functional Manager usually has more authority than the Project Manager in a functional orientate organisations. Manages Project Leader, systems architecture, technical design and delivery of solutions. Planning and managing implementation / integration of solutions. Overseeing and assisting with quality assurance testing and deployment of new systems. Providing leadership for a team of employee and consultant resources. Usually provides the cost and effort estimation of Software project.

Business Advisor

Individuals whom the best that can describe the business requirements. They are the business owners or business users. Examples of business advisers are human resources, financial, material, Purchasing, marketing managers and employees, etc, intellectual or intangible).

***Technical* Advisor/ System Analyst:**

Responsible for gathering and analyzing the business requirements and translated them to function requirements. They produce the technical Design and requirements. They are responsible for designing the hardware and software architecture. Coordinate with business advisor. Usually provides the cost and effort estimation of Software project.

Programmers/ web Developer

Translate the business and technical requirements to fully function system/application. They are expert to specific languages or tool such as Java, C++ or Oracle, or .Net programmers. Conduct the technical testing before providing the solutions to the client. Testing and debugging new or revamped computer systems and the networks on which they communicate.

Database/system/infrastructure Adm.

Responsible for designing the back end and infrastructure of the system. They produce the technical architecture to provide systems with high performance, security, availability, etc. They create the back and security policy for the system and usually their involvement are being used after the application deliver.

Project Manager

Project manager has more authority then the functional manager in project orientated organisation. They have the ultimate responsibility of the success and failure of the project. They play a major role in estimating the project effort cost. They are responsible for managing all the project resources.

III. Project's Leadership characteristics

For each item, rate how well you are able to display the ability to describe. With 9 equal to high leadership characteristics and 1 is the lowest.

Low								High
1	2	3	4	5	6	7	8	9

A. Interaction and relationship with Team members and Leadership

1. Pay personal attention to teams needs, and treat team members with respect ____
2. Take responsibilities in the face of challenges ____
3. Follow objectives with passion and encourage others ____
4. Ability to help others overcome work stress in various situations ____

B. Leadership Decision-Making

1. Lead effective meetings with a focus on decision-making process ____
2. Consulting with the teams on decision making ____
3. Make decisions with freedom and creativity within reasonable boundaries ____
4. Empower others to fulfil requirements ____

C. Ability to Motivate Team members

1. Ability to motivate team members to fulfil goals / meet targets ____
2. Share goals and visions_____
3. Provide the appropriate levels to others ____
4. Use rewards and encouragement to achieve desired results ____

D. Ability to understand the Project and organisation's culture

1. Understand and actively work to enhance cultural understanding and respect among others ____
2. Ability to effectively articulate the mission (or vision) with clarity and respect ____
3. Demonstrate the ability to understand and manage intercultural teams ____
4. Belief that everyone should conform to the majority ____

E. Active thinking

1. Develop clear vision of the project ____
2. Set achievable targets ____
3. See visions, values, and goals of the team are in line with upper management visions ____
4. Enhance creativities and teams contributions ____

F. Communication Skills

1. Communicate objectives clearly ____
2. Communicate regularly with team members ____
3. Communicate effectively for understanding, buy-in, and empowerment during the change process ____
4. Create positive atmosphere in the work environment ____

IV. Project Team Culture

For each item, rate how well you are able to display the ability to describe. With 9 equal to high characteristics and 1 is the lowest.

Low								High
1	2	3	4	5	6	7	8	9

A. Timely - attitude toward time

1. Manage time and resources in an effective manner ____
2. Respect time and value it ____
3. Understand and adapt to team's general view of time (Schedule Events vs. .Personal Relationship) ____
4. Maintain balance between work and social life ____

B. Collaboration - team relationship (impersonal relations)

1. Demonstrate a balance between personal and professional roles and responsibilities ____
2. Work effectively understanding self and others ____
3. Provide consistent feedbacks among others regarding team performance ____
4. Work with other to enhance team commitment and collaboration ____

C. Job stability – reflects loyalty

1. Create a culture of collaboration and trust so that teams take risks and achieve performance goals ____
2. Enhance team loyalty and belonging ____
3. Create trust among the different team members ____
4. Create mutual trust among themselves ____

D. Intercultural Intelligence

1. Show understanding of their shared culture ____
2. Ability to understand other cultural world views ____
3. Recognise the strengths, talents, and abilities of team when assigning roles and responsibilities ____
4. Understanding one's own feelings, values and culturally express themselves effectively ____

E. Reward Mechanism

1. Encourages and supports team in on-going professional growth opportunities ____
2. Recognise and reward the team for achievements ____
3. Use rewards and encouragement to achieve desired results ____

F. Communications

1. Use active listening and effective feedback skills ____
2. Understands and use effective relationship strategies to maintain trust within the team ____
3. Encourages the team to use their own leadership, judgement and decision–making capabilities ____
4. Talking to everyone and using teamwork to get things done ____

G. Team Experience

1. Team uses department's tools and equipment effectively ____
2. Team uses skills and knowledge acquired for the project ____
3. Team members are familiar with other team backgrounds ____
4. The organisation has pervious experience for this project ____

APPENDIX D

Exploratory Analysis of the Data

D.1 Introduction

The aim of this appendix is to produce a statistical analysis to ascertain how cultural and leadership factors affect cost estimation. This examines the importance that leaders place on cultural characteristics when it comes to project success. To gain an understanding of software measurement (and ultimately an understanding of the representation of the model), it is important to examine software characteristics in order to assess software maturity and to achieve confirmed software effort estimation improvement. As the objective of this research is to investigate the impact of various attributes on the effort, this Appendix is also an attempt to build regression models for predicting effort based on important project attributes, including cultural and leadership factors.

The process of collecting information was conducted in three stages, each using a different survey (see Chapter 3). The first survey, in which eleven government departments were selected as a Pilot Study, was conducted in Abu Dhabi. The survey was then modified and more attributes were added based on the results of the Pilot Study. This posed research questions as to whether and how culture and leadership factors have an impact on the accuracy of software effort and cost estimation. The survey results indicated that the respondents from all IT departments in the UAE concur with the significance of each of the cultural aspects covered by the survey in carrying out effort estimation. This shows that community, organisations and team members are intertwined. They are unable to escape their culture or background; they all affect each other.

This appendix is organised as follows: Section D.2 describes the statistical methodology. Section D.3 describes the distribution of actual effort. Section D.4 explores the correlation of various attributes with the actual effort. Section D.5 presents descriptive statistics for leadership and cultural characteristics, and regression analysis for estimating actual effort is discussed in Section D.6. Section D.7 provides a discussion of the statistical results. A summary is given in Section D.8.

D.2 Statistical Methodology

The original dataset consisted of survey results from 41 software projects. It was compiled after consultation with private and governmental agencies in the UAE. Some of the aforementioned attributes had missing values. These attributes were not considered in the present analysis which takes into account attributes with no or even very few missing values. Since the objective of the study is to investigate the impact of various attributes on the actual effort, the distribution of the effort needs to be examined. Two projects were incomplete, so their efforts were not recorded. Another project had an extremely large value and was removed from the analysis as an outlier. The remaining 38 projects are analysed below. Data of software attribute values must be analysed with care, because software measures are not usually normally distributed.

Several statistical techniques were used to analyse the data. Descriptive and graphical methods were used to explore and describe the values of attributes according to the type of project and organisation. A student's t-test and a one-way analysis of variance (ANOVA) were used to determine significant differences in project attributes according to type of project and organisation. Associations between attributes were assessed using Pearson's correlation when data were quantitative and the Chi-square test of independence when data were qualitative. Regression analysis was used to develop an equation for explaining the relationship between actual effort and software project attributes. The Kolmogorov-Smirnov test was used to assess the normality of data.

D.3 Distribution of Actual Effort

The distribution of 'actual effort' is highly skewed and not normally distributed. Moreover, there is an outlier (see Figure D.1). For this reason a logarithmic transformation was applied as is quite common in software cost estimation literature (Angelis et al., 2001).

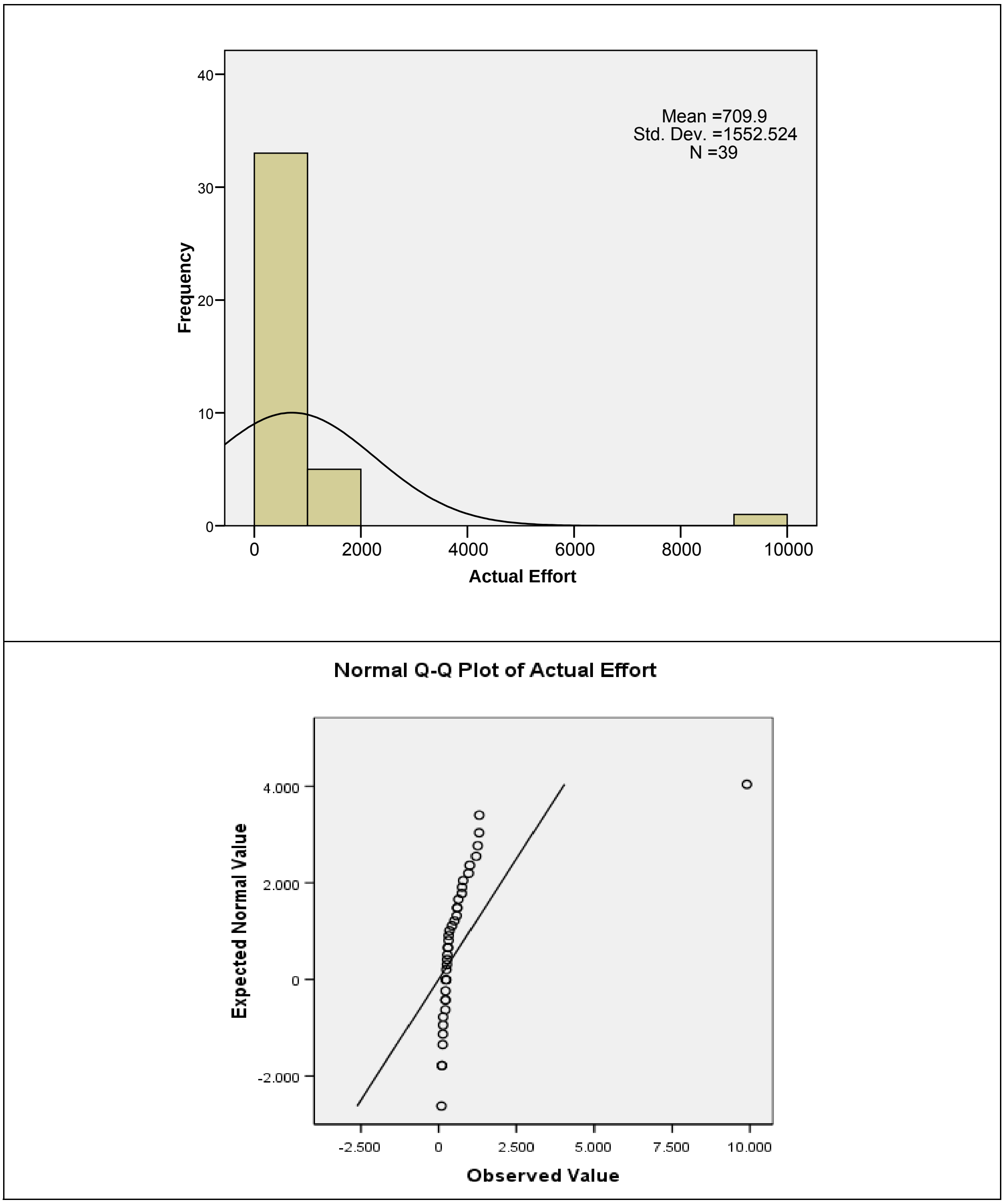


Figure D.1: The distribution of actual effort

Clearly, there is an outlier that may affect the analysis which is why it was removed from the data. As seen from the histogram and the Q-Q plot (see Figure. D.2), the distribution of the logarithm of effort can be considered as normal. The normal assumption was formally tested with the Kolmogorov-Smirnov test and indeed the distribution of the sample was found to be not significantly different from the normal (p=0.612).

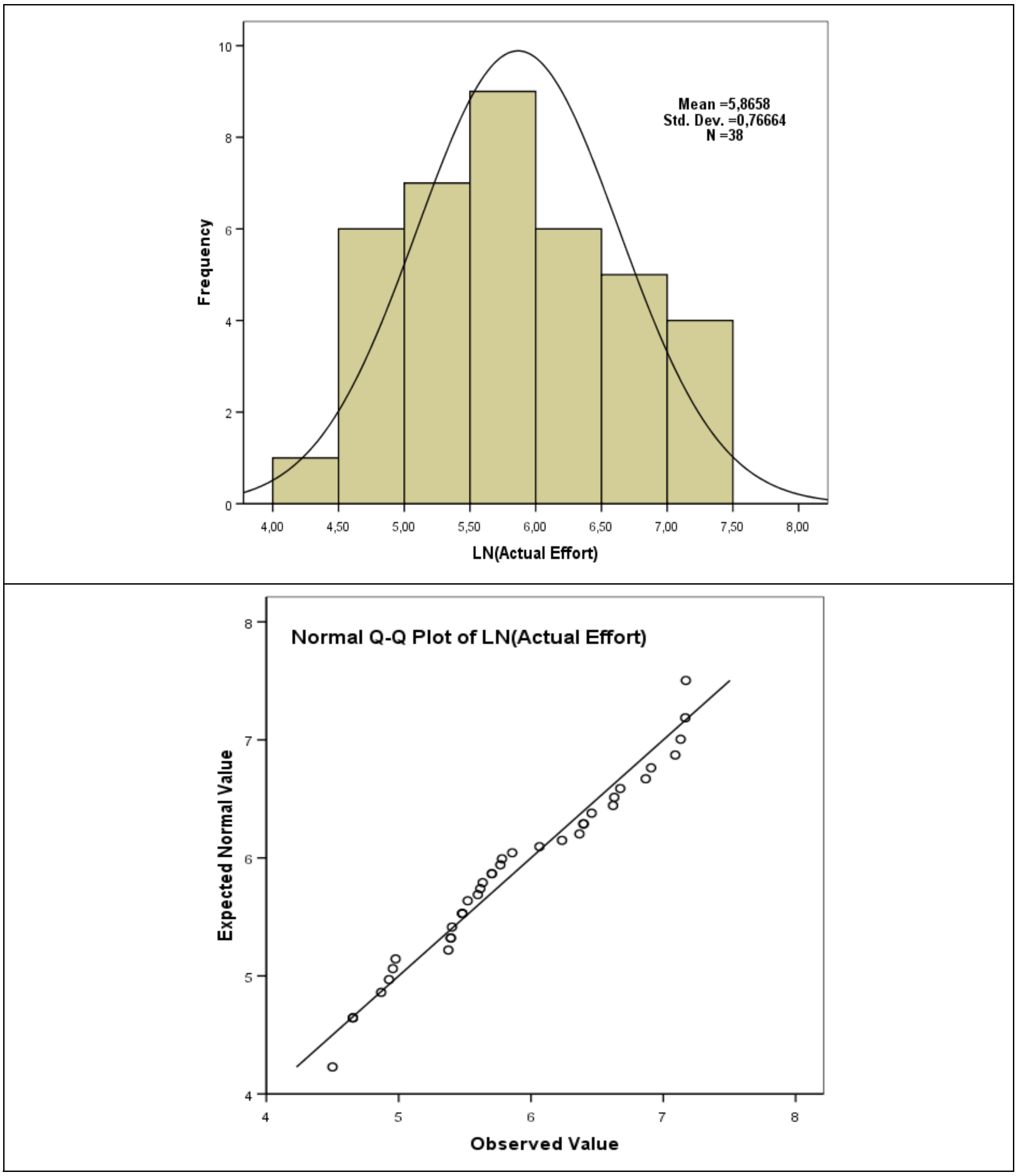


Figure D.2: The distribution of logarithm of actual effort

Although the variable effort is useful for building statistical models for predicting the cost of a new project, it is known that instead of predicting a single effort value (point estimation), it is often more useful to estimate the interval in which the true cost will fall (interval estimation). The intervals can be predefined in the sense that an analyst may define some effort categories and will then try to predict the effort category of a new project. This categorisation is often useful for finding associations with other categorical variables or for building alternative probabilistic models (like the ordinal regression in Sentas et al., 2005). For these reasons, the actual effort of developed software projects was divided into four categories based on the quartiles of its sample distribution. The values of the new ordinal variables are 1, 2, 3 and 4 and correspond to the following effort intervals (after some rounding):

Low (1) : ≤220,

Nominal (2) : (220, 300],

High (3) : (300, 638],

Very High (4) : >638

The statistics of the actual effort in each of these categories can be seen in Table D.1.

		No of Projects	Minimum	Maximum	Mean	Median	Std Deviation
Effort Intervals	Low	10	90	220	151	140	50
	Nominal	10	222	300	262	260	27
	High	9	320	638	484	510	130
	Very High	9	750	1300	1034	1000	233

Table D.1: Categories of actual effort intervals

D.4 Correlation of other attributes with the Effort

In this subsection, the correlation of the various attributes with effort was explored.

D.4.1 Organisation Size

First, a correlation analysis was applied to the original variables of effort and organisation size (number of employees) and also to their logarithms. Initially, no apparent correlation was found but a descriptive statistical analysis revealed a grouping of the organisations in two groups which seem to differ with respect to effort. For this reason, a binary variable was created dividing the organisations into two groups (see Figure D.3):

- Small (40-200 employees)
- Large (1200-16000 employees).

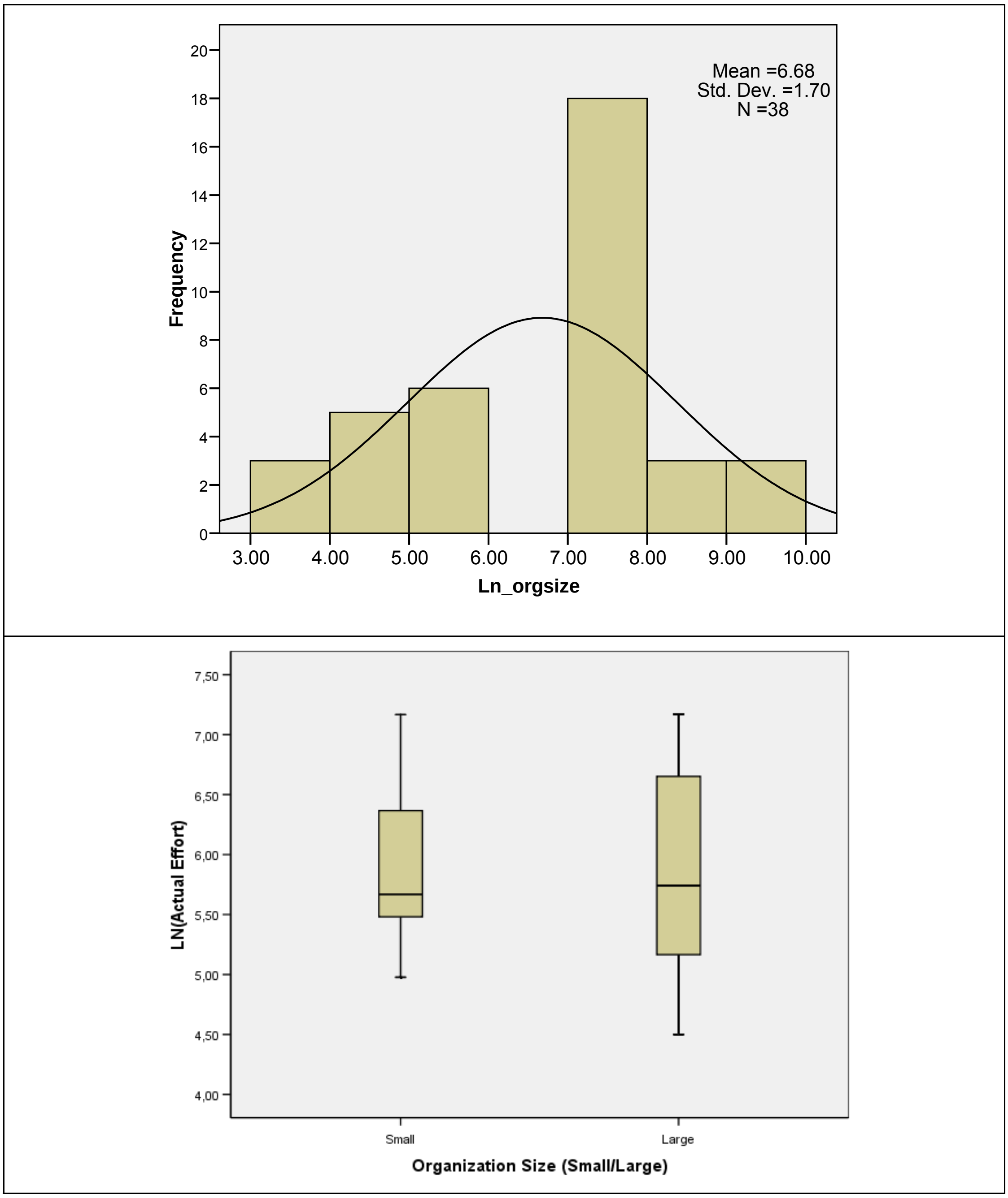


Figure D.3: Organisation size (small / large)

By creating a contingency table for the four effort categories (see Table D.1) and the categories of the organisation size and applying a Chi-Square test, the following results were obtained (see Table D.2):

			Effort Intervals				Total
			Low	Nominal	High	Very High	
Organization Size	Small	Count	1	7	5	1	14
		percentage	7.1%	50.0%	35.7%	7.1%	100.0%
	Large	Count	9	3	4	8	24
		percentage	37.5%	12.5%	16.7%	33.3%	100.0%
Total		Count	10	10	9	9	38
		percentage	26.3%	26.3%	23.7%	23.7%	100.0%

Table D.2: Distribution of actual effort levels by organisation size

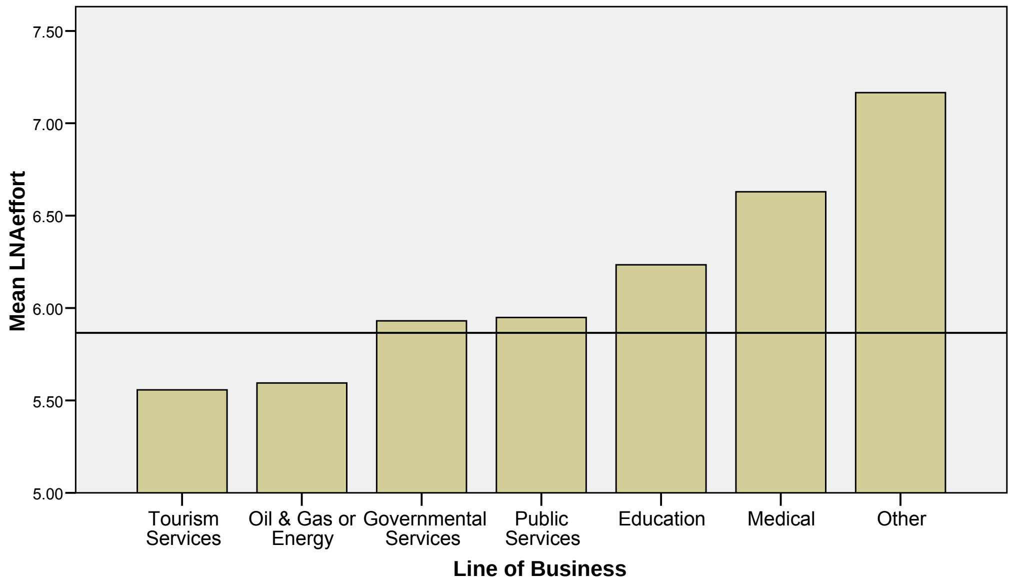
It was seen that in small organisations, projects with low and very high effort are very few; in contrast to large organisations where these two categories are the majority. This significant difference in the distributions is shown by the Chi-Square test (p=0.008).

D.4.2 Line of Business

The Line of Business is a nominal variable and contains seven categories, some of them corresponding to only one project. It was decided to group them according to effort. The mean of LN(effort) in each category is shown in Figure D.4 while the horizontal line shows the overall mean. Based on this, two categories can be defined: the first contains projects with relatively small effort (Tourist Services and Oil and Gas or Energy) and the second, all the others (Government Services, Public Services, Education, Medical and others). The t-test was applied to the LN(effort) between these groups and showed a difference between the two (p=0.071<0.10).

LN(Actual Effort)

	N	Mean	Minimum	Maximum
Medical	1	6.6294	6.63	6.63
Governmental Services	17	5.9308	4.50	7.17
Public Services	4	5.9489	4.96	6.67
Tourism Services	2	5.5577	5.48	5.63
Education	1	6.2344	6.23	6.23
Oil & Gas or Energy	12	5.5947	4.65	6.91
Other	1	7.1663	7.17	7.17
Total	38	5.8658	4.50	7.17



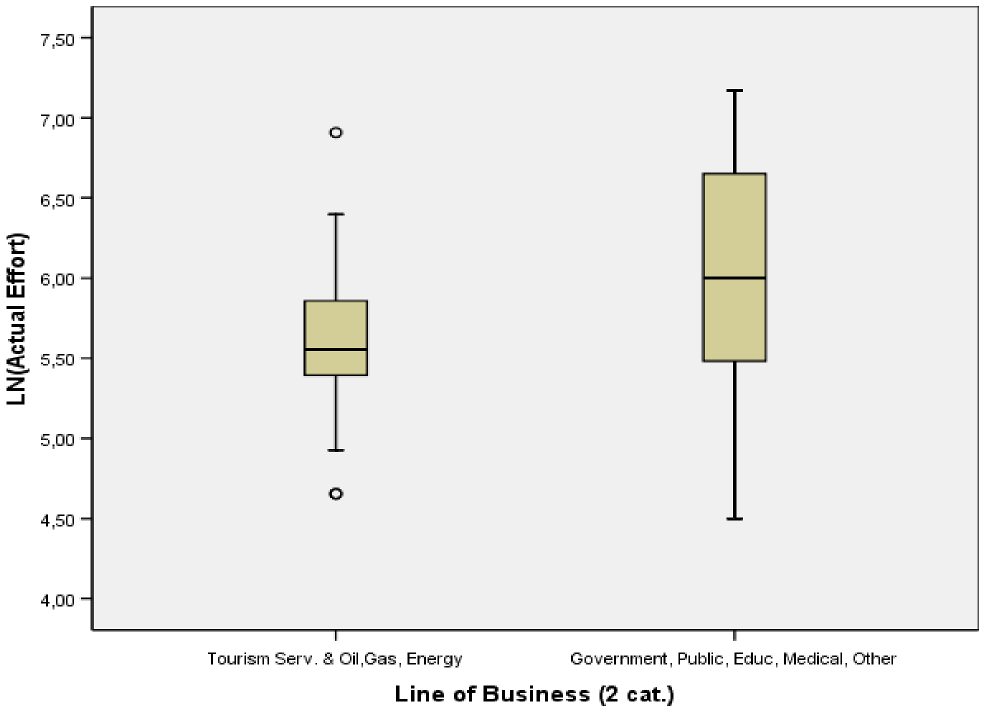


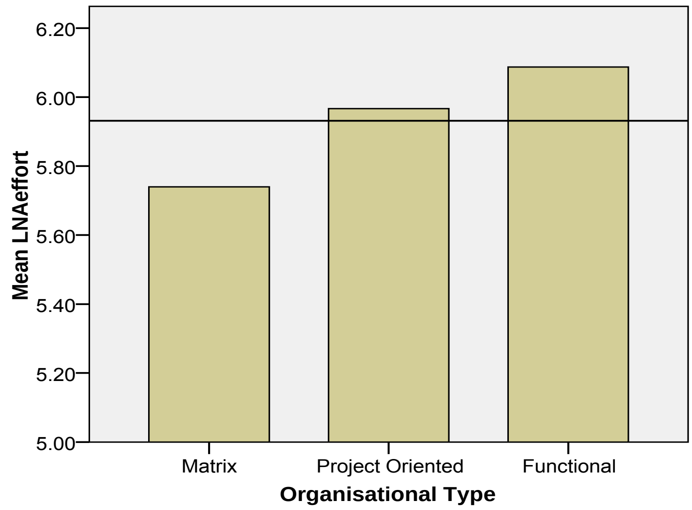
Figure D.4: The mean LN actual effort by line of business

D.4.3 Organisational Type

Organisational type initially contained three categories: 'Matrix', 'Project Oriented' and 'Functional'. A one-way ANOVA test was applied to the LN(effort) with respect to the three groups but no significant difference (p=0.587) was found. However, in order to include the variable in subsequent regression analysis, the two categories were merged with high effort (Project Oriented and Functional) in order to create a binary variable (see Figure D.5).

LN(Actual Effort)

	N	Mean	Minimum	Maximum
Project Oriented	15	5.9666	4.87	7.17
Matrix	19	5.7396	4.50	7.09
Functional	4	6.0873	5.48	7.17
Total	38	5.8658	4.50	7.17



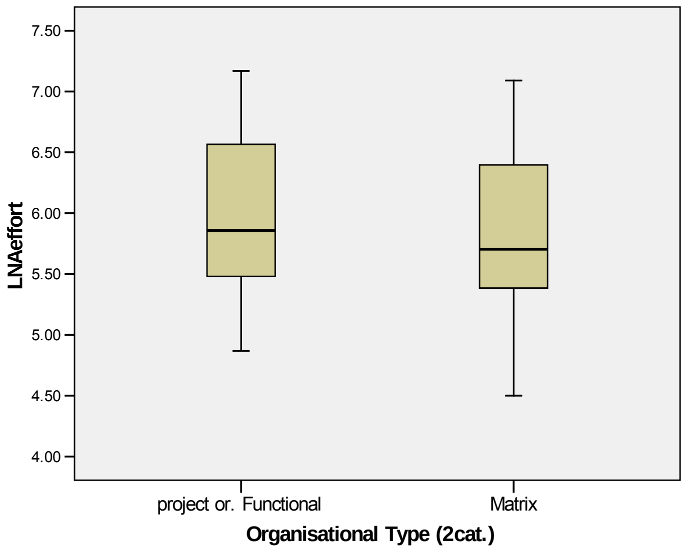


Figure D.5: The mean LN actual effort by organisational type

D.4.4 Organisation

The organisation variable initially had three categories: ‘Public-no profit’, ‘Private-profit’ and ‘Semi-government’. A one-way ANOVA test for the LN(effort) did not show a significant difference (p=0.211), although the private organisations show a increased mean of LN(effort) (see Figure D.6). Also, a binary variable was created here after merging the 'Semi-Government' and 'Public-no profit' in one category.

Descriptives

LNAeffort

Organisation	N	Mean	Minimum	Maximum
Public - no profit	32	5.8314	4.50	7.17
Private (profit)	2	6.7816	6.40	7.17
Semi-government	4	5.6831	4.87	6.67
Total	38	5.8658	4.50	7.17

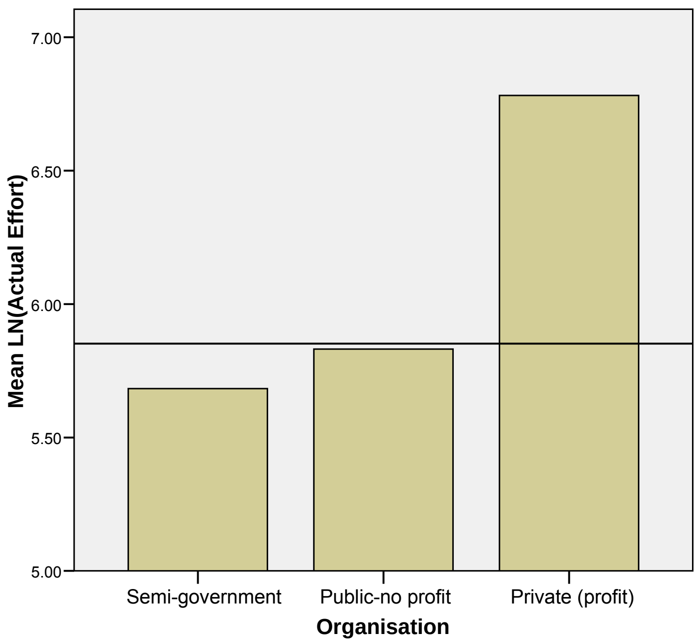


Figure D.6: The mean LN actual effort by organisation

D.4.5 Application Type

There are two types of applications in the data: ‘core’ and ‘support’. A t-test applied to LN(effort) (see Figure. D.7) did not show a significant difference (p=0.867).

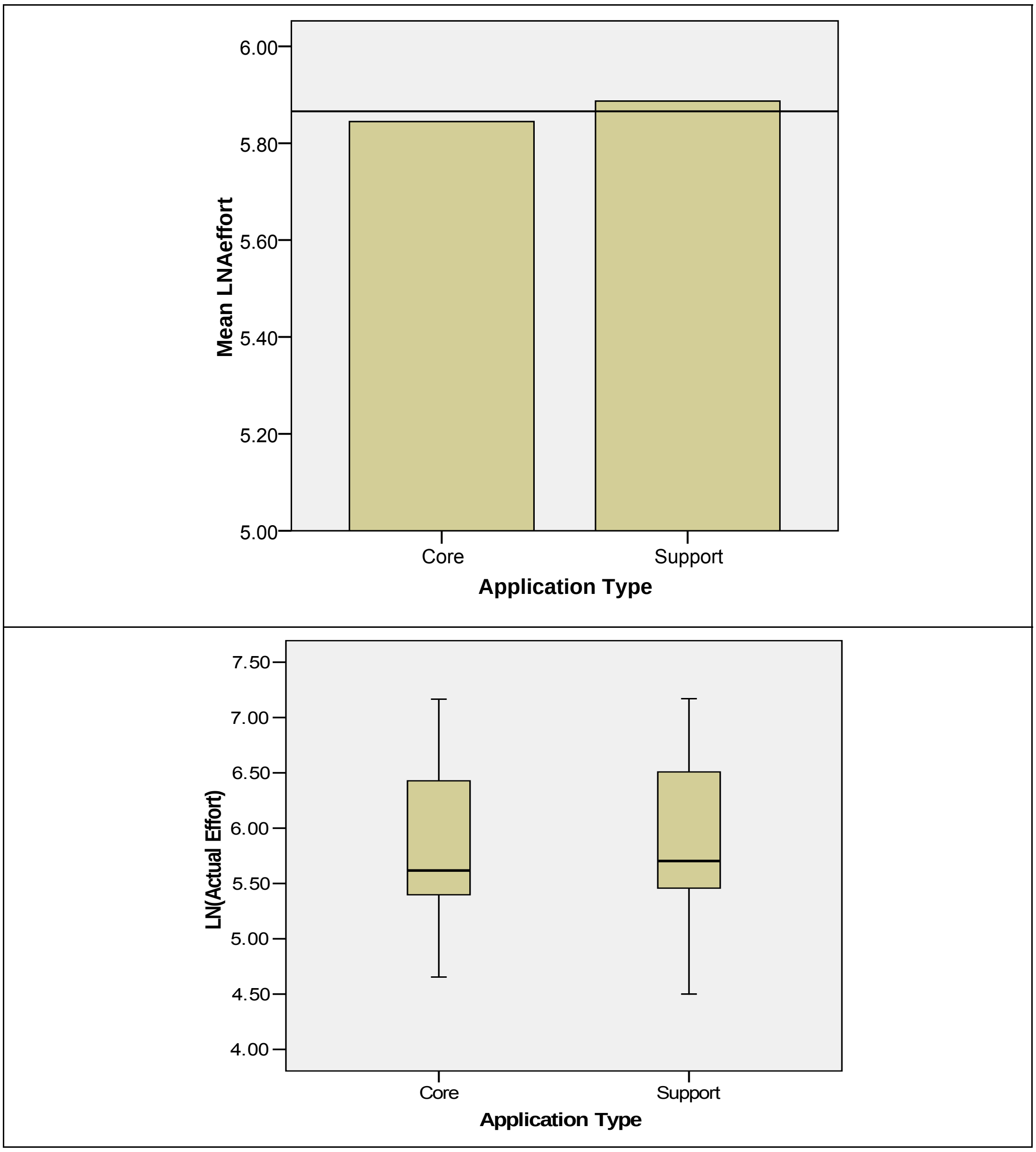


Figure D.7: The mean LN actual effort by application type

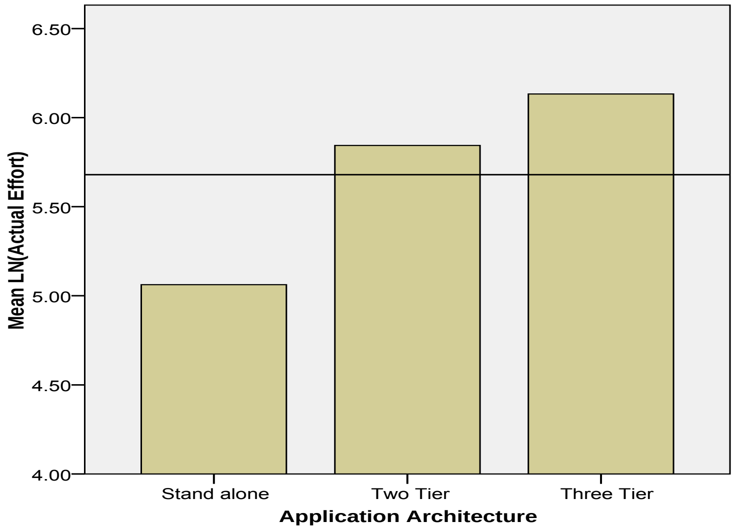
D.4.6 Application Architecture

There were three categories: ‘Stand alone’, ‘Two tier’ and ‘Three tier’ (see Figure D.8). An ANOVA test gave some indications of difference (p=0.096). A binary variable was created by the categorisation: 'stand alone, two tier’ and ‘three tier'. The subsequent t-test for the new categorisation showed a significant difference (p=0.009) between the two.

Descriptives

LN(Actual Effort)

	N	Mean	Minimum	Maximum
Stand alone	3	5.0628	4.87	5.39
Two Tier	24	5.8439	4.50	7.17
Three Tier	11	6.1326	4.98	7.17
Total	38	5.8658	4.50	7.17



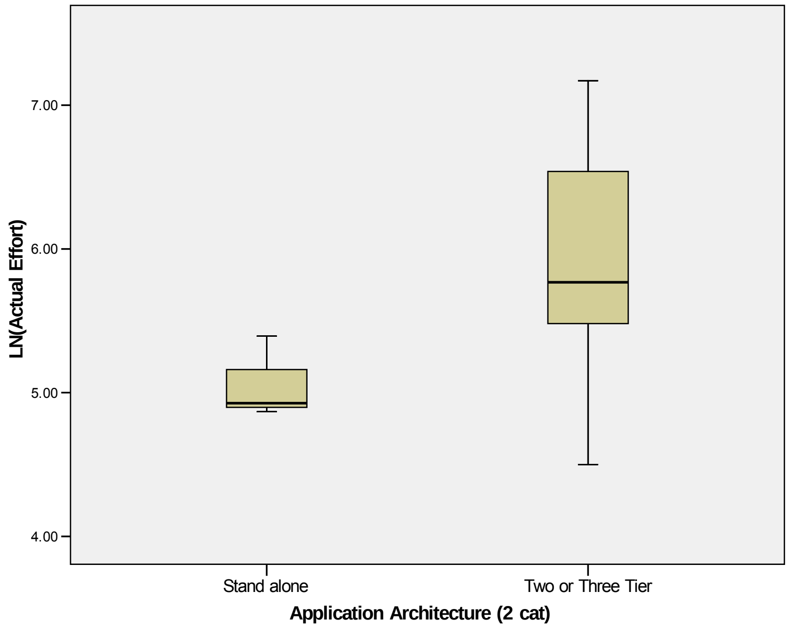


Figure D.8: The mean LN actual effort by application architecture

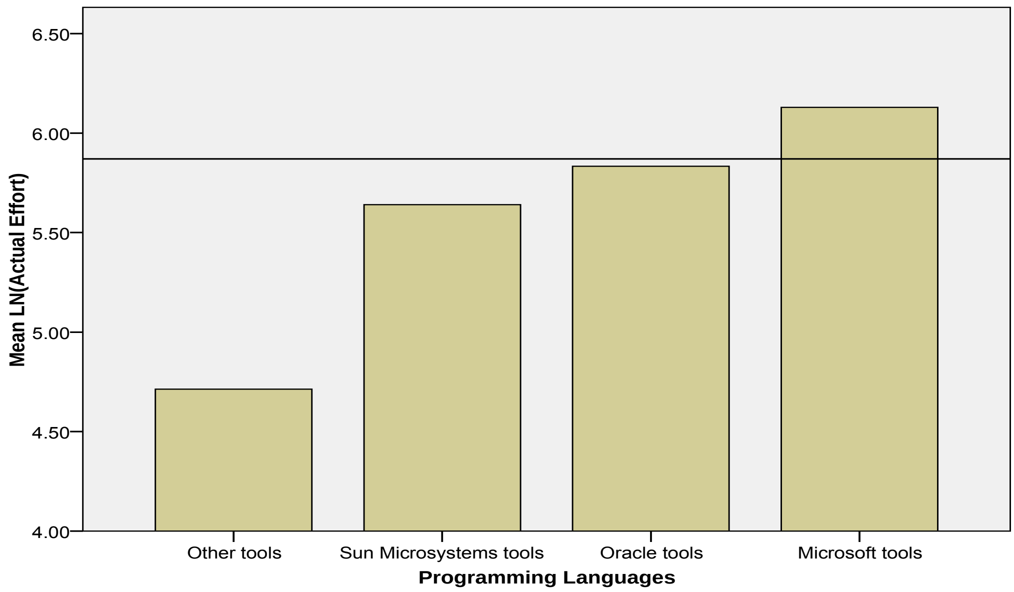
D.4.7 Programming Languages

The categories ‘Microsoft tools’, ‘Oracle tools’, ‘Sun Microsystems tools’ and ‘Other tools’ (see Figure D.9), initially showed some difference using ANOVA (p=0.074) analysis. After merging all the categories except 'Microsoft tools', the t-test produced p=0.087, indicating a difference.

Descriptives

LN(Actual Effort)

	N	Mean	Minimum	Maximum
Microsoft tools	15	6.1293	5.48	7.17
Oracle tools	16	5.8334	4.65	7.17
MicroSystems Sun tools	5	5.6403	4.65	6.91
Other tools	2	4.7135	4.50	4.93
Total	38	5.8658	4.50	7.17



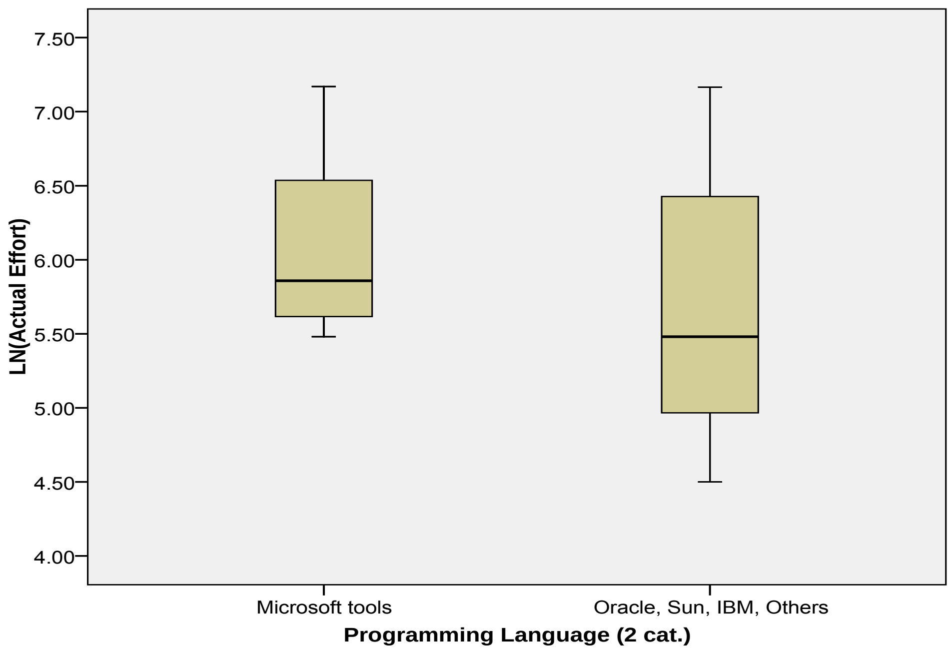


Figure D.9: The mean LN actual effort by programming languages

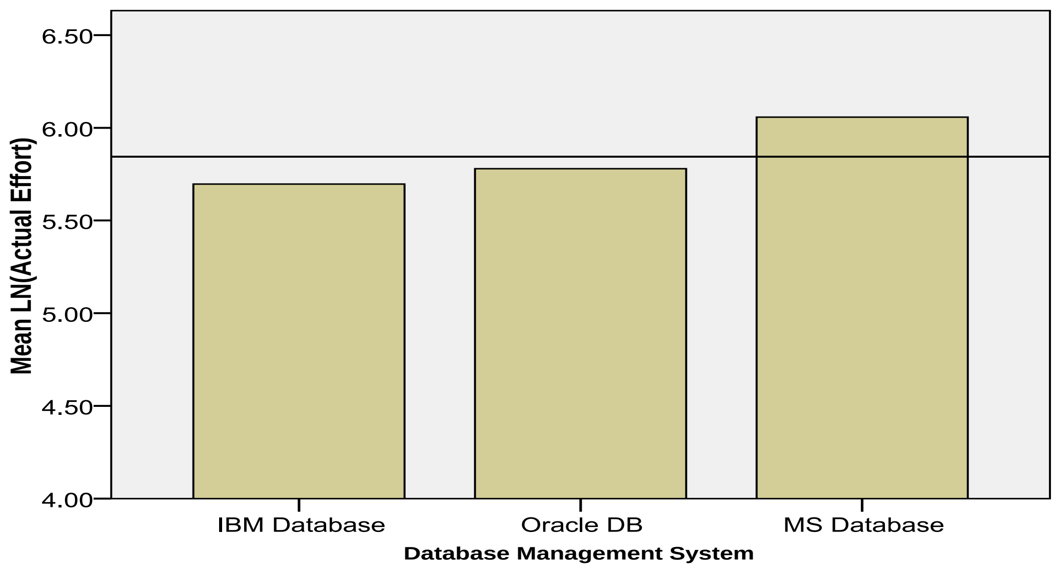
D.4.8 Database Management Systems (DBMS)

The initial categories 'MS Database', 'Oracle DB' and 'IBM Database' (see Figure D10) did not show any difference in LN(effort) by ANOVA (p=0.542). A binary variable was created by merging the categories 'IBM Database' and 'Oracle Database'.

Descriptives

LN(Actual Effort)

	N	Mean	Minimum	Maximum
MS Database	13	6.0573	4.50	7.17
Oracle DB	21	5.7796	4.65	7.17
IBM Database	4	5.6963	4.65	6.87
Total	38	5.8658	4.50	7.17



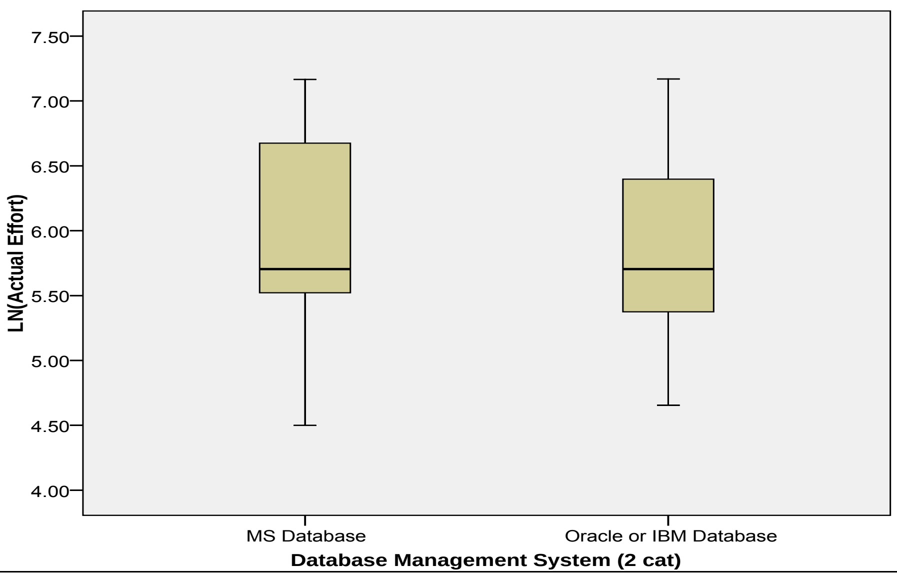


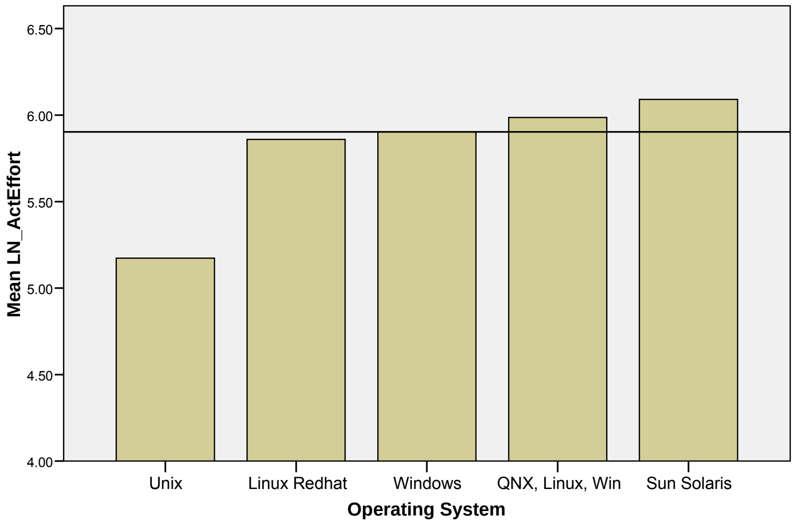
Figure D.10: The mean LN actual effort by DBMS

D.4.9 Operating Systems

The categories 'MS Windows', 'Unix', 'Linux Redhat', 'Sun Solaris' and 'QNX' (see Figure D.11) did not show any significant differences using ANOVA (p=0.593). A binary variable was produced after merging the categories 'Unix' and 'Linux Redhat' in one category and all the others in another category.

LN(Actual Effort)

	N	Mean	Minimum	Maximum
Windows	23	5.9026	4.50	7.17
Unix	3	5.1736	4.93	5.62
Linux Redhat	5	5.8596	4.65	7.17
Sun Solaris	4	6.0907	5.48	6.87
QNX, Linux, Win	3	5.9862	4.65	6.91
Total	38	5.8658	4.50	7.17



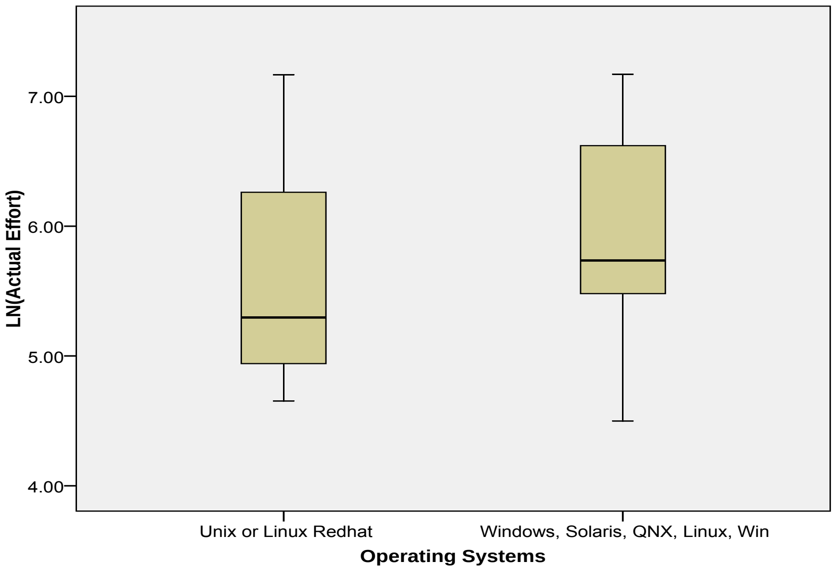


Figure D.11: The mean LN actual effort by operating systems

Table D.3 shows some summary statistics for LN(effort) for each of these project attributes. These attributes except for leadership and culture, were represented by binary variables with two categories (having values 0 and 1) in order to use them in regression models. The introduction in the same regression model, of cultural and leadership characteristics, together with other important variables and their interaction explains most of the effort variability. The mean values of LN(effort) for the different project attributes are similar with values ranging from 5.06 to 6.78, where the lowest corresponds to “Stand alone application” and the highest corresponds to “Private organisations”

			Minimum	Maximum	Mean	Median	Std Deviation	Count
Organization Size (Small/Large)	Small	LN(Actual Effort)	4.98	7.17	5.88	5.67	.57	15
	Large		4.50	7.17	5.86	5.74	.87	25
Line of Business (2 cat.)	Tourism Serv. & Oil,Gas,	LN(Actual Effort)	4.65	6.91	5.59	5.56	.62	15
	Government, Public, Educ, Medical, Other		4.50	7.17	6.03	6.00	.81	25
Organizational Type (2 cat.)	Matrix	LN(Actual Effort)	4.50	7.09	5.74	5.70	.77	20
	Proj. Or., Functional		4.87	7.17	5.99	5.86	.77	20
Organization (2cat)	Public, Semi-Government	LN(Actual Effort)	4.50	7.17	5.81	5.67	.75	37
	Private (profit)		6.40	7.17	6.78	6.78	.54	3
Application Type	Core	LN(Actual Effort)	4.65	7.17	5.84	5.62	.80	21
	Support		4.50	7.17	5.89	5.70	.76	19
Application Architecture (2 cat)	Stand alone	LN(Actual Effort)	4.87	5.39	5.06	4.93	.29	3
	Two or Three Tier		4.50	7.17	5.93	5.77	.76	37
Programming Language (2 cat.)	Oracle, MicroSyst. Sun,	LN(Actual Effort)	4.50	7.17	5.69	5.48	.81	25
	Microsoft tools		5.48	7.17	6.13	5.86	.63	15
DataBase Management System (2 cat)	Oracle or IBM Database	LN(Actual Effort)	4.65	7.17	5.77	5.70	.74	27
	MS Database		4.50	7.17	6.06	5.70	.81	13
Operating System (2 cat)	Unix or Linux Redhat	LN(Actual Effort)	4.65	7.17	5.60	5.30	.89	8
	Windows, Solaris,	LN(Actual Effort)	4.50	7.17	5.94	5.74	.73	32

Table D.3: Summary of actual effort with other attributes

D.5 The Leadership and Cultural Characteristics

Descriptive statistics for the aforementioned leadership and culture variables are shown below (see Tables D.4 and D.5).

	Mean	Median	Std. Deviation	Minimum	Maximum
Interaction and Relationship	6.9500	7.0000	1.2015	2.5000	9.0000
Decision-Making	6.8950	7.0000	1.2564	3.8000	9.0000
Ability to Motivate	6.5250	7.0000	1.3985	3.5000	8.8000
Understanding Project Culture	6.7150	7.0000	1.3552	2.8000	8.3000
Active Thinking	7.2450	7.5000	1.1507	2.0000	8.5000
Communication skills	7.2150	7.6500	1.0890	3.8000	9.0000

Table D.4: Leadership characteristics

	Mean	Median	Maximum	Minimum	Std Deviation
Timeliness	6.6658	7.0000	8.5000	1.7500	1.4575
Collaboration	6.9750	7.0000	8.5000	3.0000	1.3447
Job Stability	6.7303	7.0000	8.7500	3.2500	1.2058
Intercultural Intelligence	7.0539	7.1250	8.2500	5.0000	1.0335
Reward Mechanism	6.4298	7.0000	8.0000	2.3333	1.2566
Communication	7.1184	7.5000	9.0000	3.5000	1.3187
Team Experience	6.6461	7.0000	8.2500	3.8000	1.1155

Table D.5: Team cultural characteristics

In general, the means and the medians of all the leadership and cultural characteristics are quite high. The results were based on the survey assessment of how effective the team and leadership were in respect to the above. Regarding the correlations between the leadership characteristics, Table D.6 shows that all these characteristics correlate highly ($p<0.001$). The cultural characteristics also show strong correlations (see Table D.7). The only exception is 'Team Experience' which seems to correlate only with 'Reward Mechanism' and 'Communications'. It is also interesting to see the correlation between leadership and cultural characteristics (see Table D.8). All of the characteristics correlate highly. 'Team Experience' correlates only with 'Decision-Making' and 'Communication Skills'.

Some leadership and cultural characteristics appear to be more important than others. These characteristics were believed by the respondents to be significant attributes in most cases. This is probably due to the fact that these are innate attributes which are part of the individuals' characters which have been shaped by interaction with others and by life experience in the community.

		Interaction and Relationships	Decision Making	Ability to Motivate	Understanding Organisation Culture	Active Thinking	Communication skills
Interaction and Relationships	Pearson Correlation	1	.743**	.677**	.760**	.750**	.789**
	Sig. (2-tailed)		.000	.000	.000	.000	.000
Decision-Making	Pearson Correlation	.743**	1	.714**	.717**	.743**	.884**
	Sig. (2-tailed)	.000		.000	.000	.000	.000
Ability to Motivate	Pearson Correlation	.677**	.714**	1	.779**	.607**	.645**
	Sig. (2-tailed)	.000	.000		.000	.000	.000
Understanding Organisation Culture	Pearson Correlation	.760**	.717**	.779**	1	.688**	.648**
	Sig. (2-tailed)	.000	.000	.000		.000	.000
Active Thinking	Pearson Correlation	.750**	.743**	.607**	.688**	1	.793**
	Sig. (2-tailed)	.000	.000	.000	.000		.000
Communication skills	Pearson Correlation	.789**	.884**	.645**	.648**	.793**	1
	Sig. (2-tailed)	.000	.000	.000	.000	.000	

**. Correlation is significant at the 0.01 level (2-tailed).

Table D.6: Correlation between leadership characteristics

		Timeliness	Collaboration	Job Stability	Intercultural Intelligence	Reward Mechanism	Communication	Team Experience
Timeliness	Pearson Correlation	1	.700**	.791**	.693**	.520**	.598**	.286
	Sig. (2-tailed)		.000	.000	.000	.001	.000	.073
Collaboration	Pearson Correlation	.700**	1	.694**	.667**	.686**	.666**	.170
	Sig. (2-tailed)	.000		.000	.000	.000	.000	.295
Job Stability	Pearson Correlation	.791**	.694**	1	.752**	.449**	.476**	.133
	Sig. (2-tailed)	.000	.000		.000	.004	.002	.413
Intercultural Intelligence	Pearson Correlation	.693**	.667**	.752**	1	.443**	.453**	.110
	Sig. (2-tailed)	.000	.000	.000		.004	.003	.498
Reward Mechanism	Pearson Correlation	.520**	.686**	.449**	.443**	1	.584**	.303
	Sig. (2-tailed)	.001	.000	.004	.004		.000	.057
Communication	Pearson Correlation	.598**	.666**	.476**	.453**	.584**	1	.467**
	Sig. (2-tailed)	.000	.000	.002	.003	.000		.002
Team Experience	Pearson Correlation	.286	.170	.133	.110	.303	.467**	1
	Sig. (2-tailed)	.073	.295	.413	.498	.057	.002	

**. Correlation is significant at the 0.01 level (2-tailed).

Table D.7: Correlation between culture characteristics

		Timeliness	Collaboration (Interpersonal Relation)	Job Stability	Intercultural Intelligence	Reward Mechanism	Communication	Team Experience
Interaction and Relationships	Pearson Correlation	.692**	.690**	.738**	.571**	.458**	.585**	.165
	Sig. (2-tailed)	.000	.000	.000	.000	.004	.000	.322
Decision-Making	Pearson Correlation	.677**	.849**	.628**	.529**	.571**	.706**	.338*
	Sig. (2-tailed)	.000	.000	.000	.001	.000	.000	.038
Ability to Motivate	Pearson Correlation	.598**	.790**	.610**	.472**	.666**	.487**	.092
	Sig. (2-tailed)	.000	.000	.000	.003	.000	.002	.583
Understanding Organisation Culture	Pearson Correlation	.714**	.773**	.741**	.644**	.690**	.449**	.132
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.005	.430
Active Thinking	Pearson Correlation	.609**	.750**	.558**	.398**	.538**	.591**	.242
	Sig. (2-tailed)	.000	.000	.000	.013	.000	.000	.144
Communication skills	Pearson Correlation	.580**	.776**	.501**	.469**	.611**	.701**	.345*
	Sig. (2-tailed)	.000	.000	.001	.003	.000	.000	.034

**. Correlation is significant at the 0.01 level (2-tailed). *.Correlation is significant at the 0.05 level (2-tailed).

Table D.8: Correlation between culture and leadership characteristics

D.6 Regression Analysis

Using all the binary variables from Section D.4 and all the leadership and cultural characteristics of Section D.5, the construction of a regression model with the most important variables was attempted. The purpose of this analysis was to see whether and how the various characteristics, and especially 'Leadership' and 'Culture', have an effect on the effort. After experimenting with different combinations, the following models were produced showing the coefficients (and therefore the corresponding variables) to be significant at the 0.10 level.

D.6.1 Model 1

Using all these binary variables and the leadership and cultural characteristics, the following model resulted (see Table D.9).

Coefficients[a]

Model		Unstandardized Coefficients B	Unstandardized Coefficients Std. Error	Standardized Coefficients Beta	t	Sig.
1	(Constant)	4.998	.922		5.422	.000
	Organization Size (Small/Large)	.658	.253	.420	2.597	.014
	Organizational Type (2 cat.)	.702	.236	.464	2.972	.006
	Organization (2cat)	1.319	.476	.389	2.769	.009
	Application Architecture (2 cat)	1.138	.421	.406	2.704	.011
	Operating System (2 cat)	.508	.255	.274	1.990	.056
	Team Experience	-.213	.099	-.310	-2.155	.039

a. Dependent Variable: LN(Actual Effort)

LN(effort) = 4.998 + 0.658**Organisation Size (small/large)* + 0.702**Organisational Type (2 cat)* + 1.319 * *Organisation (2 cat)* + 1.138**Application Architecture (2cat)* + 0.508**Operating System (2cat)* – 0.213**Team Experience*

Table D.9: Regression model 1

The model accounts only for the 46.9% of the variation of the dependent variable (logarithm of effort) since $r^2 = 0.469$ while the adjusted r^2 is 0.366. The F-test shows that the model is statistically significant (p=0.002) and the residual analysis (see Figure D.12) shows that the distribution of the residuals is not significantly different from the normal (the Kolmogorov-Smirnov test gives p=0.493). Also, only one standardised residual is greater than 2.

The fitting accuracy of the model is computed from the estimated effort (by the model) and then by taking the magnitude of relative error (MRE).

$$mre = \left| \frac{actual\_effort - estimated\_effort}{actual\_effort} \right|$$

Then the MMRE and the pred25 are calculated by:

$$mmre = mean(mre)\,,\ mdmre = median(mre)\,,\ pred25 = \frac{\#mre < 0.25}{\#mre}$$

Here the fitting accuracy measures are MMRE = 0.5046 (or 50.46%), MdMRE = 0.3474 (or 34.74%) and pred25=0.3157 (31.57%). Regarding the predictive accuracy of the model, the adjusted predicted values were computed, i.e. the predicted values of each case when the specific case is excluded from the calculation of the regression coefficients. The measures of predictive accuracy are MMRE = 70%, MdMRE = 40% and pred25 = 16%.

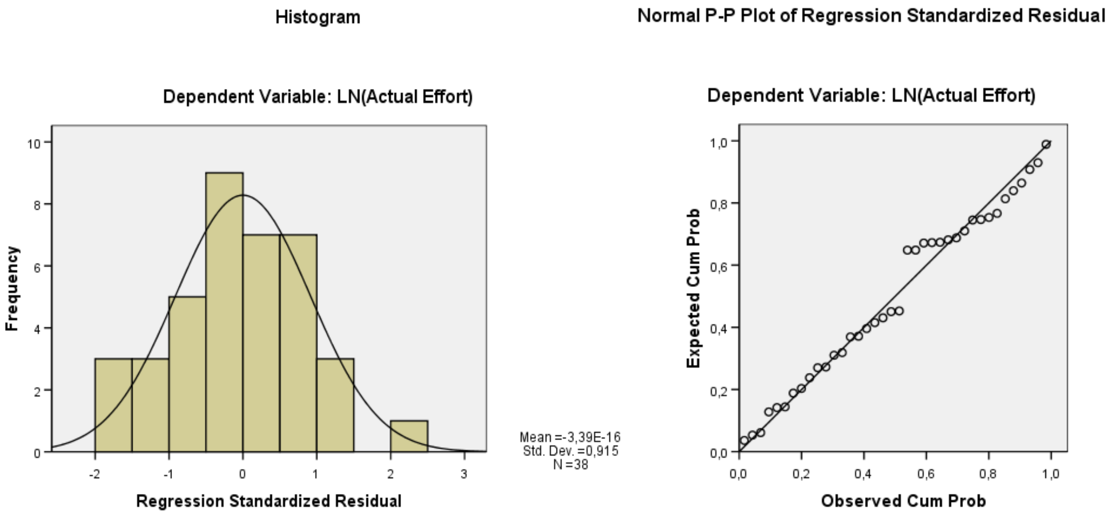


Figure D.12: Distribution of the residuals of model 1

Obviously, the model's accuracy is quite low despite its significance. An interesting point is that the cultural characteristic 'Team Experience' participates in the model with a negative sign, i.e. the high values of this characteristic are associated with low effort values. Some of the other characteristics can also enter in the equation, but the high correlations between them make the models unstable and unexplainable although the validation measures are better (the phenomenon of multicollinearity). This can be shown by the following model.

D.6.2 Model 2

This model is significant (p=0.002) and all the independent variables are significant too at the 0.10 level. Also, $r^2 = 0.595$ and adjusted r^2 is 0.445. The residuals are normally distributed. The fitting accuracy measures are MMRE = 39.87%, MdMRE = 26.36% and pred25 = 50%. The predicting accuracy measures are MMRE = 64.19%, MdMRE = 38.92% and pred25 = 42.10%. However, the positive coefficients of “Timeliness” and “Communications” are not intuitively explained (see Table D.10).

LN(effort) = 6.161+ 0.976 * Organisation size(small/large) + 0.409 * Line of Business
+ 0.690 * Organisational type (2cat) + 0.892 * Organisation (2cat)
+ 1.223 * Application architecture (2cat) – 0.249 * Active thinking
- 0.309 * Intercultural intelligence - 0.303 * Team experience + 0.270 * Timeliness
+ 0.209 * Communications

Table D.10: Regression model 2

D.6.3 Model 3

In this model, the logarithm of function points was used. This has some missing values, so only 29 projects were used to build the following model (see Table D.11).

LN(effort) =	5.546 + 0.145 * ln(UFP) + 1.080 * Organisation size(small/large) + 0.554 * Organisational type (2cat) +2.177 * Organisation (2cat) + 1.432 * Application architecture (2cat) - 0.482 * Data Base Management System + 0.438 * Decision-making - 0.265 * Motivation - 0.299 * Interpersonal Relation + 0.348 * Reward mechanism - 0.623 * Team experience

Table D.11: Regression model 3

The model is significant (p= 0.000<0.0005) and all the independent variables are significant too at the 0.10 level. Also, $r^2 = 0.817$ and adjusted r^2 is 0.699. The residuals are normally distributed. The fitting accuracy measures are MMRE = 24.14%, MdMRE = 18.07% and pred25 = 68.97%. The predicting accuracy measures are MMRE = 48.21%%, MdMRE = 27.44% and pred25 = 46.42%. Again, despite the good performance, the model suffers from intuitive interpretability due to the signs of certain variables.

D.7 Discussion of the Statistical Results

Apart from the statistical analysis presented, other methods were also tried on the data, focusing especially on the possibility of generating cost estimation models from all the variables and especially the leadership and cultural characteristics. Specifically:

- A cluster analysis of all the leadership and cultural characteristics revealed the existence of two clusters in the data, one with generally low values of the items and the other with high values. This is in accordance with the high correlation found between all the items. The variable was then used in regression analysis but without significant improvements.
- The four effort intervals defined in Section D.3 were used as an ordinal dependent variable for an ordinal regression (OR) model (Sentas et al., 2005) with predictors of all the variables discussed in the previous

sections. These models are very informative when they are fitted to the data but they are also quite complicated. In this case, the OR models do not seem to contribute greatly to the understanding of the impact of leadership and cultural characteristics on the effort.

- Estimation by Analogy was also applied along with the jack-knife procedure to see whether the predictions were improved. However, no significant improvement was attained.
- A significant point that deserves further investigation is the impact of leadership and cultural characteristics on the accurate prediction of the effort by the organisation. Specifically, the initial estimation of the effort was registered in this data (as it was predicted by the organisation itself along with the actual effort), so, it was possible to calculate the magnitude of relative error (MRE) of each project and therefore the relative error each organisation made in its own prediction. What was found, although not statistically proved, is that the majority of the higher errors belonged to the cluster of projects with low leadership and cultural characteristics. This suggests that when the values of such characteristics are high, the schedule of the project is more easily followed. However, as already mentioned, this conjecture needs serious investigation with new data.

Generally speaking, the statistical analysis provides valuable indications which can be further exploited in the continuance of this research. In this respect, a number of problems in the data, which made the building of a sound model difficult, should be mentioned. First of all, there were organisations which participated, each with 4-5 projects, which were quite different in effort but more or less the same in their leadership and cultural characteristics. This was a significant source of variation affecting the model accuracy. The second problem, as seen in Section D.5., was the high correlation of all the leadership and cultural characteristics. So the existence of only one of these variables in a model is an indication of a more general effect but without an efficient expression.

These are interesting results to note but they did not help the construction of a cost model. The problem is that the effort is essentially uncorrelated with all attributes and that regression is not a good way to handle such data. Therefore, the direction now is: a) to work only with analogy; b) to work with total skill cost which appears to be correlated with other variables; c) to split the data in core/ support which may help as there is a significant difference in their mean LNeffort.

D.8 Summary

Several statistical tests and techniques were presented to analyse the data attributes. Culture and leadership have been recognised as important attributes affecting cost estimation models. The proposed models can be applied to other cultures. The data analysis is important to identify patterns and relationships which help form judgements about the attributes being measured. This work is an elaboration of the survey work done in Chapter 3. The aim of the survey was to find out how cultural features affect the cost estimation models.

In order to measure these crucial human characteristics objectively and accurately, reliability is needed throughout software development rather than at the end of the process. These measurements may prove valuable in the building of effort prediction models.

APPENDIX E

The SEEOS Data Definition and Data Dictionary

1) SEEOS Data definition

Field Name	*Data Type*	*Size*	*Remark*
User_ID	Text	4	Key
Password	Password	8	Data

Organisation User Table Definition: Design View

User_ID	*Password*
UØØ1	********
UØØ2	*******
UØØ3	*******
UØØ4	*******

ORGANISATION USER TABLE: Data view

Field Name	*Data Type*	*Size*	*Remark*
OrgNo	Text	4	Key
BUS_ID	Text	3	Key
Culture_ID	Text	3	Key
User_ID	Text	4	Key
OrgName	Text	30	Data
Org_Size (# of Employee)	Number	Long Integer	Data

Organisation Table Definition: Design View

OrgNo	*Bus_ID*	*Culture_ID*	*User_ID*	*OrgName*	*OrgSize*
OØØ1	BØ1	CØ1	UØØ1	AFD	*4000*
OØØ2	BØ2	CØ2	UØØ2	ADG	*1300*
OØØ3	BØ3	CØ3	UØØ3	ZAD	*2140*
OØØ4	BØ4	CØ4	UØØ4	ADC	*2295*
OØØ5	BØ5	CØ5	UØØ5	UAE	*2000*
OØØ6	BØ6	CØ6	UØØ6	GIA	*83*
OØØ7	BØ1	CØ1	UØØ7	AMU	*1650*
OØØ8	BØ2	CØ2	UØØ8	APN	*2800*
OØØ9	BØ3	CØ3	UØØ9	PWD	*2000*
OØ10	BØ4	CØ4	UØ10	ADC	*2500*
OØ11	BØ5	CØ5	UØ11	ADW	*550*
OØ12	BØ6	CØ6	UØ12	PGH	*2000*
OØ13	BØ1	CØ7	UØ13	CIV	*1000*
OØ14	BØ2	CØ2	UØ14	JUS	*1000*
OØ15	BØ3	CØ3	UØ15	CAV	*2000*
OØ16	BØ4	CØ4	UØ16	DNR	*1000*
OØ17	BØ5	CØ5	UØ17	DHM	*8100*
OØ18	BØ6	CØ6	UØ18	EGV	*60*
OØ19	BØ5	CØ5	UØ19	DMU	*150*
OØ20	BØ6	CØ6	*UØ20*	AEC	*1500*

ORGANISATION TABLE: Data view

Field Name	*Data Type*	*Size*	*Remark*
OrgNo	Text	4	*Key*
Project_ID	Text	4	*Key*
SYS_MGT_ID	Text	3	*Key*
DEV_ID	Text	3	*Key*
Project_Type	Text	100	*Data*
Start_Date	Date/Time	dd-mm-yy	*Data*
Finish_Date	Date/Time	dd-mm-yy	*Data*
Estimation Cost	Currency	--	*Data*
Actual Cost	Currency	--	*Data*
Application_ID	Text	3	*Foreign Key*
Leadership_ID	Text	3	*Foreign Key*
Actual Effort	Number	Long Integer	*Data*

Project Table Definition: Design View

OrgNo	*Project ID*	*SYS_MGT ID*	*DEV _ID*	*Project Type*	*Start Date*	*Finish Date*	*Est. Cost*	*Actual Cost*	*App ID*	*LID*	*Estimate Effort*	*Actual Effort*
OØØ1	PØØ1	MØ1	DØ1	HR	9/2000	7/2002	2 M	3.5M	AØ1	AØ1	500	580
OØØ2	PØØ2	MØ2	DØ2	FS	7/2002	7/2002	1.5 M	1.9 M	AØ2	AØ2	600	700
OØØ3	PØØ3	MØ3	DØ3	ERP	7/2002	9/2005	9.5 M	15 M	AØ1	AØ1	952	952
OØØ4	*PØØ4*	*MØ1*	*DØ4*	*EDMS*	*9/2005*	--	*6.5 M*	*7 M*	*AØ2*	*AØ2*	*1497*	*1650*

Project Table: Data view

Field Name	*Data Type*	*Size*	*Remark*
Application _ID	Text	3	Key
Application _Type	Text	10	Data
Business Users	Number	Long Integer	Data
# of Clients	Number	Long Integer	Data
# of Transaction	*Number*	*Long Integer*	*Data*

Project Business Environment Table Definition: Design View

Application ID	*Application Type*	*Business Users (# Backend Users)*	*# of Clients*	*# of Transactions*
AØ1	Support	100	100	*50*
AØ2	Core	200	200	*20*

Project Business Environment Table: Data view

Field Name	*Data Type*	*Size*	*Remark*
BUS_ID	Text	3	Key
Busines_Type (Domain)	Text	35	Data

Business Type Table Definition: Design View

Bus_ID	*Business Type (Domain)*
BØ1	*Medical*
BØ2	*Governmental Services*
BØ3	*Communication*
BØ4	*Public Services*
BØ5	*Tourism Services*
BØ6	*Education*
BØ7	*Oil &Gas*

BUSINESS TYPE TABLE: Data view

Field Name	*Data Type*	*Size*	*Remark*
SYS_MGT_ID	Text	3	*Key*
SYS_MGT_Type	*Text*	*35*	*Data*

System Management Type Table Definition: Design View

SYS_MGT_ID	*SYS_MGT_Type*
MØ1	*Project Oriented (Projectorized)*
MØ2	*Functional (low authority)*
MØ3	*Matrix (intermediate)*

SYSTEM MANAGEMENT TYPE TABLE: Data view

Field Name	*Data Type*	*Size*	*Remark*
OrgNo	Text	4	*Key*
Project_ID	Text	4	*Key*
Entity_Code	Text	3	*key*
Measurement	*Number*	*Long Integer*	*Data*

Project Entity Size Table Definition: Design View

OrgNo	*Project No*	*Entity code*	*Measurement (Size)*
O001	P001	EØ1	5000
O002	P002	EØ2	50
O003	P003	EØ3	30
O004	P004	EØ4	5
O005	P005	EØ5	35
O006	P006	EØ6	20
O007	P007	EØ7	25

PROJECT ENTITY SIZE TABLE: Data view

Field Name	*Data Type*	*Size*	*Remark*
Entity_Code	Text	3	*key*
Entity_Description	*Text*	*40*	*Data*

Entity Table Definition: Design View

Entity code	*Entity Type (Description)*
EØ1	Line of Code (LOC)
EØ2	UML-DFD
EØ3	FUNCTION POINTS
EØ4	Entity Relationship Diagram (ERD)
EØ5	Object Diagram (Classes, Attributes)
EØ6	Structure Chart (Process)
EØ7	Tables/Forms/Reports

ENTITY TABLE: Data view

Field Name	*Data Type*	*Size*	*Remark*
SKL_ID	Text	3	*Key*
Skill_Role (Type)	Text	15	*Data*
EXP_Years	*Number*	*Long Integer*	*Data*

Skill Levels Table Definition: Design View

SKL_ID	*Skill Role (Type)*	*EXPerience_Years*
SØ1	Junior	<5
SØ2	Senior	*5-10*
SØ3	Principal	>10

SKILL LEVELS TABLE: Data view

Field Name	*Data Type*	*Size*	*Remark*
OrgNo	Text	4	*Key*
Project_ID	Text	4	*Key*
Skill_ID	Text	3	*Key*
SKL_ID	Text	3	*Key*
Tech_ID	Text	3	*Key*
Region_ID	Text	3	*Key*
Skill_Cost	Currency		*Data*
Estimate_man_days	Number	Long Integer	*Data*
Actual_man_days	*Number*	*Long Integer*	*Data*

Skill Cost Table Definition: Design View

OrgNo	*Project_ID*	*Skill_ID*	*SKL_ID*	*Tech_ID*	*Region_ID*	*Skill_Cost**	*Est_ days months*	*Act_days month*
OØØ1	PØØ1	KØ1	SØ1	TØ1	RØ1	$75.00	6	9
OØØ2	PØØ2	KØ2	SØ2	TØ2	RØ2	$150.00	24	27
OØØ3	PØØ3	KØ5	SØ3	TØ3	RØ3	$250.00	12	28
OØ20	PØØ4	KØ6	SØ4	TØ4	RØ4	$500.00	8	6

SKILL COST TABLE: Data view

Field Name	*Data Type*	*Size*	*Remark*
Skill_ID	Text	3	*Key*
Skill_Name	*Text*	*40*	*Data*

Skill Type Table Definition: Design View

Skill_ID	*Skill Name*
KØ1	*Database Development and Adm.*
KØ2	*System Analysis*
KØ3	*Network Design*
KØ4	*Programming/Software Eng.*
KØ5	*Technical Advisor*
KØ6	*Web Development*
KØ7	*Application Manager*
KØ8	*IT Consultant*
KØ9	*IT Project Manager*

SKILL TYPE TABLE: Data view

Field Name	*Data Type*	*Size*	*Remark*
DEV_ID	Text	3	*Key*
Development_Type	*Text*	*20*	*Data*

Development Type Table Definition: Design View

DEV_ID	*Development_Type*
DØ1	*New System*
DØ2	*Major Upgrade*
DØ3	*Minor Upgrade*

DEVELOPMENT TYPE TABLE: Data view

Field Name	*Data Type*	*Size*	*Remark*
OrgNo	Text	4	*Key*
Project_ID	Text	4	*Key*
Tech_ID	Text	3	*key*
Tech _Cost	*Currency*		*Data*

Project Technical Table Definition: Design View

OrgNo	*Project _ID*	*Tech_ID*	*Tech_Cost (total)*
O001	P001	TØ1	4000
O002	P002	TØ2	2000
O003	P003	TØ3	1000
O004	P004	TØ4	7000
O005	P005	TØ5	5000
O006	P006	TØ1	1000

Project Technical Table: Data view

Field Name	*Data Type*	*Size*	*Remark*
Tech_ID	Text	3	*Key*
Generation	Text	20	*Data*
Tech_Description	Text	20	*Data*
Tech_Used	Text	30	*Data*
Tech_Unit_Cost	*Number*	*Double*	*Data*

PROJECT Technical Environment Definition: Design View

Tech_ID	*Generation*	*Technical Description*	*Technology used Hardware & Software Infrastructures*	*Technical Unit Cost*
TØ1	4th G	Java ORACLE C++ MY SQL	Win Based	*1000*
TØ2	3rd G	C FORTRAN VB	Unix Based	*700*
TØ3	2nd G	Access EXCEL GUI	Apple X	*300*
TØ4	*Applications*	*Assembler* *DB2*	*Main Frame*	*100*

PROJECT Technical Environment TABLE: Data view

Field Name	*Data Type*	*Size*	*Remark*
Region_ID	Text	3	*Key*
Region	Number	%	*Data*
GDP	Number	%	*Data*
Inflation Rate	*Number*	%	*Data*

Regional Table Definition: Design View

Region_ID	*Region*	*Life_Expenses GDP**	*Inflation_Rate**
RØ1	Asia	19.7 pct; 5.2 Growth	4.8 pct
RØ2	Middle east	28.8 pct; 5.4 Growth	1.85 pct – 1.69 pct
RØ3	Europe	38.3 pct; 2.7 Growth	2.43 pct -2.45 pct
RØ4	South East Asia	5.1 pct;5.4 pct	4.9 pct – 5.2 pct
RØ5	East & North Asia	6.4 pct – 6.6 pct	2.2 pct -2.3 pct
RØ6	South Africa	--	-1.5 pct – -0.5 pct
RØ7	North America	20.6 pct– 2.7 Growth	2.1 pct – 2.5 pct
RØ8	South America	2.6 Growth	8.7 pct – 9.8 pct

Regional TABLE: Data view

*http://www.forbes.com/markets/feeds

Field Name	*Data Type*	*Size*	*Remark*
CULT_ID	Text	3	*Key*
Cult_Description	Text	20	*Data*
Timely	Number	Integer	*Data*
Collaboration Team Effort	Number	Integer	*Data*
Stability	Number	Integer	*Data*
Communication Structure	Number	Integer	*Data*
Intercultural Intelligence	Number	Integer	*Data*
Reward	Number	Integer	*Data*
Team Experience	Number	Integer	*Data*

Culture Table Definition: Design View

CID	*(Timely) Respect Time*	*Collaboration Team Effort*	*Job Stability*	*Communication Structure*	*Intercultural Intelligence (knowledge skill)*	*Reward (Incentives)*	*Team Experience*
CØ1	High (7-9)	High (7-9)	High (7-9)	High (7-9)	High (7-9)	High (7-9)	High (7-9)
CØ2	Medium (4-6)	Medium (4-6)	Medium (4-6)	Medium (4-6)	Medium (4-6)	Medium (4-6)	Medium (4-6)
CØ3	Low (1-3)	Low (1-3)	Low (1-3)	Low (1-3)	Low (1-3)	Low (1-3)	Low (1-3)

CULTURE TABLE: Data view

Field Name	*Data Type*	*Size*	*Remark*
Leadership_ID	Text	3	*Key*
Leadership_Type	Text	20	*Data*
Interaction & Relationship	Number	Integer	*Data*
Decision-Making	Number	Integer	*Data*
Motivation	Number	Integer	*Data*
Understanding Org. Culture	Number	Integer	*Data*
Active Thinking	Number	Integer	*Data*
Communication Skills	*Number*	*Integer*	*Data*

Project Leadership Table Definition: Design View

LID	*Interaction*	*Decision Making*	*Motivation*	*Understanding*	*Active Thinking*	*Communication Skills*
L01	High (7-9)	High (7-9)	High (7-9)	High (7-9)	High (7-9)	High (7-9)
L02	Medium (4-6)	Medium (4-6)	Medium (4-6)	Medium (4-6)	Medium (4-6)	Medium (4-6)
L03	Low (1-3)	Low (1-3)	Low (1-3)	Low (1-3)	Low (1-3)	Low (1-3)

Project Leadership TABLE: Data view

2) SEEOS Data Dictionary

Field Name	*Field Description*	*Data Type*	*Field Size*	*Sample Data*	*Usage*
Technical Cost	Each Project Technical has assigned of technical total cost ✓ TØ1 ✓ TØ2	Text	3	2000 1000	1. Project Technical Table 2. Report 3. Entry
Business Users	Each Project Technical has assigned of Backend users	Number	Long Integer	100 ERP users	1. Project Business Environment Table 2. Culture Table 3. Report 4. Entry
# of Clients	Each Project Number of Clients has assigned of Business Environment	Number	Long Integer	10 20 50	1. Project Business Environment Table 2. Report 3. Entry
# of Transactions	Each Project has assigned of Number of Transaction for each Project Environment, transactions by Weekly ✓ ≤1000 ✓ 1001<TRN<10,000 ✓ 10,001<TRN<500,000 ✓ >500,000	Number	Long Integer	200 100	1. Project Business Environment Table 2. Report 3. Entry

Field Name	*Field Description*	*Data Type*	*Field Size*	*Sample Data*	*Usage*
Region_ID	Each Culture has assigned of Regional ID key ✓ RØ1 ✓ RØ2 ✓ RØ3 ✓ RØ4 ✓ RØ5 ✓ RØ6 ✓ RØ7	Text	3	R01 R04	1. Regional Table 2. Report 3. Entry
Region	Each Culture has assigned of Region Key ✓ Asia ✓ Middle east ✓ Europe ✓ South East Asia ✓ East Asia ✓ South Asia ✓ North America ✓ South America	Text	20	Middle east UK	4. Regional Table 5. Report 6. Entry
Cost of Living GDP+	Each Region has assigned of Regional Life Expense percentage for last 5-10 years ✓ 4.1 pct ✓ 2.1 pct	Number	Long Integer	5%	7. Regional Table 8. Report 9. Entry
Inflation_Rate+	Each Region has assigned of regional Inflation rate	Number	Long Integer	4.5%	10. Regional Table 11. Report 12. Entry
Goss/Population % of Consumer Price Index	*Each Culture has assigned of Regional CPR key*	Number	Long Integer	3%	13. Regional Table 14. Report 15. Entry

Field Name	*Field Description*	*Data Type*	*Field Size*	*Sample Data*	*Usage*
Cult_Type	Each Culture has assigned of Culture Type 9-7: High 6-4: Medium 3-1: Low	Number	Long Integer	9 7	1. Culture Table 2. Report 3. Entry
Timely	Each Culture has assigned of Timely key 9-7 High, 6-4 Medium , 3-1 Low	Number	Long Integer	9 7	1. Culture Table 2. Report 3. Entry
Collaboration Team Effort	Each Culture has assigned of Collaboration Level 9-7 High, 6-4 Medium , 3-1 Low	Number	Long Integer	9	1. Culture Table 2. Report 3. Entry
Stability	Each Culture has assigned of Stability Key 9-7 High, 6-4 Medium , 3-1 Low	Number	Long Integer	7	1. Culture Table 2. Report 3. Entry
Communication Structure	*Each Culture has assigned of Communication Structure Key* 9-7 High, 6-4 Medium, 3-1 Low	Number	Long Integer	9	1. Culture Table 2. Report 3. Entry
Intercultural Intelligence	Each Culture has assigned of Intelligence Level Key 9-7 High, 6-4 Medium, 3-1 Low	Long Integer	Number	9	1. Culture Table 2. Report 3. Entry
Team Experience	Each Team Culture has assigned of Team Experience Level Key 9-7 High, 6-4 Medium, 3-1 Low	*Number*	*Long Integer*	5	1. Culture Table 2. Report 3. Entry

Field Name	*Field Description*	*Data Type*	*Field Size*	*Sample Data*	*Usage*
Reward	Each Culture has assigned of Reward Level Key 9-7 High, 6-4 Medium , 3-1 Low	Long Integer	Number	5	1. Culture Table 2. Report 3. Entry
Evaluation	Each Culture has assigned of Evaluation Level Key. 9-7 High, 6-4 Medium , 3-1 Low	Long Integer	Number	1	1. Culture Table 2. Report 3. Entry
Leadership_Type	Each Leadership has assigned of Leadership type key 9-7 High, 6-4 Medium , 3-1 Low	Long Integer	Number	5	1. Leadership Table 2. Report 3. Entry
Interaction & Relationship	*Each Leadership has assigned of Interaction & Relationship Level key* 9-7 High, 6-4 Medium , 3-1 Low	*Long Integer*	*Number*	9	1. Leadership Table 2. Report 3. Entry

Field Name	*Field Description*	*Data Type*	*Field Size*	*Sample Data*	*Usage*
Decision-Making	Each Leadership has assigned of Decision Making Level key 9-7 High, 6-4 Medium, 3-1 Low	Long Integer	Number	9 5 1	4. Leadership Table 5. Report 6. Entry
Motivation	Each Leadership has assigned of Motivation Level key 9-7 High, 6-4 Medium, 3-1 Low	Long Integer	Number	9	4. Leadership Table 5. Report 6. Entry
Understanding Org. Culture	Each Leadership has assigned of Understanding Org. Culture key 9-7 High, 6-4 Medium, 3-1 Low	Long Integer	Number	5	4. Leadership Table 5. Report 6. Entry
Active Thinking	Each Leadership has assigned of Active Thinking Level key 9-7 High, 6-4 Medium, 3-1 Low	Long Integer	Number	1	7. Leadership Table 8. Report 9. Entry
Communication Skills	*Each Leadership has assigned of Communication Skills Level key* 9-7 High, 6-4 Medium, 3-1 Low	Long Integer	Number	9	4. Leadership Table 5. Report 6. Entry

APPENDIX F

Profile Theory Techniques

In this theory, an object is described by a set of factors and, in turn, each factor is defined by multiple characteristics. A set of factors forms a profile. A factor is represented by qualitative and quantitative information. A profile is introduced as follows (Plekhanova, 2000): Quantitative description of the ith profile factor is defined by time characteristic, property, and weight. In particular,

- ε_i - time characteristic of the i^th^ factor $\varepsilon_i = \varepsilon_i(t)$ (e.g. duration/length of experience, factor existence or non existence). Domain constraints may define bounds, i.e. $\varepsilon_i^b \le \varepsilon_i \le \varepsilon_i^u$, where $\varepsilon_i^b \ge 0$, $\varepsilon_i^u \ge 0$ represent bottom (lower) and top (upper) values of the i^{th} factor time range, respectively.

- v_i - property of the i^{th} factor (e.g. depth, level, range, complexity or capability of a factor). Since a property may change with time, $v_i = v_i(t)$ can be defined as a function of time. Domain constraints may define bounds, i.e. $v_i^b \le v_i \le v_i^u$, where $v_i^b \ge 0$, $v_i^u \ge 0$ represent bottom (lower) and top (upper) values of the i^{th} factor property range, respectively.
- w_i - weight of a factor which defines either the factor importance or the factor priority. Factor weights can vary, and therefore, w_i can be also considered as a function of time $w_i = w_i(t)$. Domain constraints may define bounds, i.e.

$w_i^b \leq w_i \leq w_i^u$, where $w_i^b \geq 0$, $w_i^u \geq 0$ represent bottom (lower) and top (upper) values of the ith factor weight range, respectively.

In order to define completeness of knowledge/skill description of a project performer with respect to the required factors for a task, a consideration of the property for completeness of profile factors is needed. That is, the set LS represents a complete set of the key profile factors if the set of the factor weights for the particular profile satisfies the following conditions:

$$\sum_{i=1}^{n} w_i(t) = 1, \text{ where } w_i(t) \in [0,\ 1]$$

The completeness property is important for the identification of the essential profile factors and for these to be incorporated into the profile description. Thus, it is possible, using a notion of profile presented in profile theory, to describe leadership. For example, factors that represent a leadership (LS) could be described as follows:

$$f(LS) = \{< \varepsilon_1, L, \omega_1) >, < \varepsilon_2, T, \omega_2 >, < \varepsilon_3, G, \omega_3 >, < \varepsilon_4, C, \omega_4 >, < \varepsilon_5, X, \omega_5 >\}$$

Where:

L : Leadership characteristics, such as style, power, capability, traits, and skills

T : Team characteristics, such as culture, knowledge, personal competencies

G : Organisational type, such as project-oriented, functional, or matrix authority

C : Communication skills, in both channels (leader vs. team culture)

X : Project complexity, such as core or support systems

ε_i : Factor existence, such as ε = 1, non existence ε = 0; where ε_i is

ε_1, ε_2, ε_3, ε_4, and ε_5 are the factor existence for leader characteristics, team culture, organisational type, communication skills, and complexity, respectively.

ω_i: Total weight of sub-factor(s) weight is divided equally in approximation ad hoc cases or based on importance or priority, where ω_1, ω_2, ω_3, ω_4, and ω_5, are the weights for leadership characteristics, team culture, organisational type, communication skills and complexity, respectively.

For example, a leader's profile may be defined by five parameters to measure the leadership factor:

$$L = \{(<\varepsilon_i, l_i, \omega_i>), i = \overline{1,n}\}, \quad n = 5$$
$$L = \{(<\varepsilon_1, l_1, \omega_1>, <\varepsilon_2, l_2, \omega_2>, <\varepsilon_3, l_3, \omega_3>, <\varepsilon_4, l_4, \omega_4>, <\varepsilon_5, l_5, \omega_5>)\}$$

For example L = <1, 7.1, 1/5>, <1,6.5, 1/5>…

The team profile may consist of six parameters such as team culture, experience, and capability:

$$T = \{(<\varepsilon_i, t_i, \omega_i>), i = \overline{1,n}\}, \quad n = 6$$
$$T = \{(<\varepsilon_1, t_1, \omega_1>, <\varepsilon_2, t_2, \omega_2>, <\varepsilon_3, t_3, \omega_3>, <\varepsilon_4, t_4, \omega_4>, <\varepsilon_5, t_5, \omega_5>, <\varepsilon_6, t_5, \omega_6>)\}$$

The organisational type (G) profile may be defined by 3 parameters

$$G = \{(<\varepsilon_i, G_i, \omega_i>)\}, G_i = \begin{cases} 0, & \text{if G is functional - project manager has less authority} \\ 0.5, & \text{if G is matrix - project manager and functional manager share similar authority} \\ 1, & \text{if G is project - oriented - project manager has the highest authority} \end{cases}$$

$$G = \{(<\varepsilon_i, G_i, \omega_i>), i = \overline{1,3}\}$$

$$G = \{(<\varepsilon_1, G_1, \omega_1>, <\varepsilon_2, G_2, \omega_2>, <\varepsilon_3, G_3, \omega_3>)\}$$

The communication skills profile may be defined as:

$$C = \{(<\varepsilon_i, c_i, \omega_i>), i = \overline{1,n}\}, \ \text{n} = 2$$

The project complexity (X) profile may be defined by 2 parameters

$$X = \{(<\varepsilon_i, X_i, \omega_i>), i = \overline{1,2}\}, X_i = \begin{cases} 0.1, & \text{if X is Support} \\ 0.2, & \text{if X is Core} \end{cases}$$

$$X = \{(<\varepsilon_1, X_1, \omega_1>, <\varepsilon_2, X_2, \omega_2>)\}$$

If weights are equal as approximated, then $\omega(f_i) = 1/n \quad , 0 \le \omega f_i \le 1/n$

Where $\omega(f_i)$ are maximum weights for each factor (i.e. 1/5 = 0.2), otherwise weights are based on factor priority or importance.

These factors are considered in isolation and need to be integrated to assess their contribution to the success of a project. The set of attributes were measured on a 1-9 scale through a questionnaire, a mean value calculated. The results of the study showed that the means of leadership and culture were very close 7.0 and 6.8 (see Table F.6). Profile theory thus allows for the assessment of a leader in a comprehensive manner including many different factors.

Projno	Factors	Interaction	Decision Making	Motivation	Undestanding Project Culture	Active Thinking	Communication skill	Average Leader	Time - Respect	Impersonal Relation Collaboration	Job Stability	Intercultural Intelligence	Reward Mechanism	Communications	Team Experience	Average culture
All	Average	7.1	7.0	6.6	6.8	7.2	7.1	7.0	6.7	6.9	6.8	7.1	6.5	7.1	6.8	6.8

Table F.6: Average score for leadership and culture attributes

Where:

v : The input value for each factor

k : The weight of each factor

s : The maximum scale value

n : The number of parameters

w : The maximum weight for each factor

For each parameter, the weighting factor is the maximum weight divided by the number of parameters $\frac{\omega(f_i)}{n} = k$; for example, $\frac{0.2}{5} = 0.04$, where 5 is number of parameters for leader factors. The output value of a factor in leadership is a set of weights, as defined as follows: $parm\_\omega = \frac{v \times k}{s}$; the weight for each parameter is based on an input value multiplied by the weight of each factor and divided by the maximum scale value. In this example (see Table F.7), the organisational hierarchy was project-oriented and the application type was support systems.

Description	Factor	No. of parm (n)	max value (s)	Parameter	Parmater Value (v)	max factor weight (w) = 1/ n	Weight each factor (k) = w / n	Paramter Weight (parm_w)
Leader	L	5.00	9.00	l1	7.1	0.20	0.040	0.03
				l2	7.0			0.03
				l3	6.6			0.03
				l4	6.8			0.03
				l5	7.2			0.03
							w(L)	0.154
Team	T	6.00	9.00	t1	6.7	0.2	0.033	0.02
				t2	6.9			0.03
				t3	6.8			0.03
				t4	7.1			0.03
				t5	6.5			0.02
				t6	6.8			0.03
							w(T)	0.151
Orgn.	G	1.00	3.00	Functional	0	0.2	k = 0 | ½ | 1	-
				Matrix	0.5			-
				Prj. Oriented	1			0.20
							w(G)	0.200
Commu. Skills	C	2.00	9.00	Lcommu.	7.1	0.2	0.10	0.08
				Tcommu.	7.1			0.08
							w(C)	0.128
Complexity	X	1.00	2.00	Support	0	0.2	k = 0 | 1	0.10
				Core	1			-
							w(X)	0.100
							Total required weight	0.67

Table F.7: Leadership profile

A profile for leadership factors is the total of all sub-factors. The value of the profile leadership (LS) total weight is equal to $\omega(LS)_i = \sum_{i=1}^{n} \frac{v \times k}{s}$, (i.e. 0.67) for this case, by adding all sub-factors for leadership gives a ratio which can be compared with other sub-factors of leadership to see which profile carries more weight in term of project success.

The following question seems appropriate: "What is more important: having an exemplary and experienced leader, or having a perfect environmental culture? Which one has more impact?" Experts believe that both are important because both have close results, but the capability and compatibility of each part is also important. In order to evaluate leadership capability, the following profile is used:

$$V(Ls_i, Cu_i) = \omega_{\mathrm{i}} \left(\frac{\varepsilon_i}{\varepsilon_i^{(0)}} \right) \left(\frac{v_i}{v_i^{(0)}} \right)^2 \qquad (1)$$

Hence, for the ith factor:

ε_i : Available or previous experience (existence)

$\varepsilon_i^{(0)}$: Required experience (existence)

v_i : Existing level capability of factors

$v_i^{(0)}$: Required level capability of factors

ω_i : Weight factor

A profile capability (V) for leadership or culture is the sum of all factor capabilities in formula (1).

$$V(Ls_i, Cu_i) = \sum_{\mathrm{i}=1}^{\mathrm{n}} \omega_{\mathrm{i}} \left(\frac{\varepsilon_i}{\varepsilon_i^{(0)}} \right) \left(\frac{v_i}{v_i^{(0)}} \right)^2 \qquad (2)$$

Leader compatibility may be defined as leader capability that is used for the project development without adaptation, adjustment or modification. In order to determine compatibility of a leader and provide comparison of different available factor/profile capabilities of a leader with respect to the required factor/profile capability of the leader, a compatibility weight can be used for the measure defined in the profile theory as follows (Plekhanova, 1999): a compatibility-weight is an integrated quality characteristic that represents compatibility of available profile capability with required profile capability. Thus, a compatibility weight for leadership is defined as follows in formula (3).

$$w(Ls_i, Cu_i) = \prod_{i=1}^{n} \left(\frac{\varepsilon_i}{\varepsilon_i^{(0)}} \right) \left(\frac{v_i}{v_i^{(0)}} \right)^2 \qquad (3)$$

where $\mathbf{w}(Ls_i, Cu_i) \geq 1$, if n available factor capabilities cover n required factor capabilities, i.e. $V(Ls_i, Cu_i) \geq V(Ls_i, Cu_i^{(0)})$, $\forall i: \; i = \overline{1, n}$.

The weights for leadership and cultural compatibility are computed by the number of required and number of available factors. Assume that a project manager A (PM A) was obtained from the average observed cases of leadership and cultural attributes. The corresponding data values were collected and considered as average cases (see Table F.6). An ordinary project manager (PM B) was compared with the 'optimum' project manager (see Tables F.8 and F.9 respectively). The required case helps in assessing new cases. A new case is evaluated and compared with a given required case. The candidate is assessed using the obtained values for the attributes. The measured capabilities and compatibilities are used to determine how close the candidate's case is to the required case or to an average case. The decision is based on closest weight.

Factor	Required -Optimum			PM A (average Cases)			PM B		
Leader	attrib	prop_level	weight =1/6	attrib	prop_level	weight =1/6	attrib	prop_level	weight =1/6
Interaction	9	3	0.17	7.1	3	0.17	6.5	3	0.17
Decision Making	9	3	0.17	7.0	3	0.17	5.3	2	0.17
Motivation	9	3	0.17	6.6	3	0.17	6.3	3	0.17
Undestanding Project Culture	9	3	0.17	6.8	3	0.17	6.8	3	0.17
Active Thinking	9	3	0.17	7.2	3	0.17	6.5	3	0.17
Communication Skills	9	3	0.17	7.1	3	0.17	6.6	3	0.17
others	0	0	0	0	0	0	0	0	0
Leader Capability	v(l)	w(l)		v(l)	w(l)		v(l)	w(l)	
v(l1)	0.17	1.00		0.13	0.79		0.12	0.72	
v(l2)	0.17	1.00		0.13	0.78		0.04	0.26	
v(l3)	0.17	1.00		0.12	0.73		0.12	0.70	
v(l4)	0.17	1.00		0.13	0.76		0.13	0.76	
v(l5)	0.17	1.00		0.13	0.80		0.12	0.72	
v(l6)	0.17	1.00		0.13	0.79		0.12	0.73	
Total v(l)	1.00			0.77			0.65		
Length	6			6			6		
p(l)-compatibility length	1.00			1.00			1.00		
w(l) - compatiblity weight	1.00			0.21			0.05		
w(l(1/5)) comp.weight	1.00			0.27			0.07		

Table F.8: Leadership factors and capability / compatibility values

In this study, six leadership characteristics were identified. These attributes were derived by averaging each sub-level of characteristic. The levels were based on the assumption of level of the leadership (high = 3). The weight was 1/6 equal priority for all factors. For example, the leader profile capability for (PM A) attribute such as 'interaction' is $v(Ls_1)$; therefore,

$v(Ls_1) = 0.17\left(\frac{7.1}{9}\right)\left(\frac{3}{3}\right)^2$; where the level scale is 1-3; low, nominal, and high respectively. The total capability $v(Ls_i)$ is the sum of all $v(Ls_i)$ where i is the number for factors. The length is the number of factors which are available or which satisfy the required profile and is positive (>0). The compatibility length $p(l)$ is the ratio of available factors over required factors. The compatibility weight $w(l)$ is an integrated quality characteristic which represents the available compatibility and the required profile compatibility. Therefore, assuming that project task requires finding (PM A), then we get: $w(l)$ is 0.79*0.78*0.73*0.76*0.8*0.79 = 21%. PM A communication was used in a separate profile with culture therefore only 5 factors were satisfied. Thus, the compatibility-weights ω(I(1÷6) = 0.79*0.78*0.73*0.76*0.8 = 27%. A similar approach was applied to the culture and skills profiles.

Factor	Required -Optimum			PM A (average Cases)			PM B		
Culture	attrib	prop_level	weight =1/7	attrib	prop_level	weight =1/7	attrib	prop_level	weight =1/7
Time - Respect	9	3	0.14	6.7	3	0.14	6.5	3	0.14
Impersonal RelationCollaboration	9	3	0.14	6.9	3	0.14	5.3	3	0.14
Job Stability	9	3	0.14	6.8	3	0.14	6.3	2	0.14
Intercultural Intelligence	9	3	0.14	7.1	3	0.14	6.8	3	0.14
Reward Mechanism	9	3	0.14	6.5	3	0.14	6.5	2	0.14
Communications	9	3	0.14	7.1	2	0.14	7	3	0.14
Team Experience	4	3	0.14	6.8	2	0.14	6	2	0.14
others	0	0	0	0	0	0	0	0	0
Culture Capability	v(c)	w(c)		v(c)	w(c)		v(c)	w(c)	
v(c1)	0.14	1.00		0.11	0.74		0.10	0.72	
v(c2)	0.14	1.00		0.11	0.77		0.08	0.59	
v(c3)	0.14	1.00		0.11	0.76		0.04	0.31	
v(c4)	0.14	1.00		0.11	0.79		0.11	0.76	
v(c5)	0.14	1.00		0.10	0.72		0.05	0.32	
v(c6)	0.14	1.00		0.05	0.35		0.11	0.78	
v(c7)	0.14	1.00		0.11	0.76		0.10	0.67	
Total v(c)	1.00			0.70			0.59		
Length	7			7			7		
p(c)-compatibility length	1.00			1.00			1.00		
w(c) - compatiblity weight	1.00			0.07			0.02		
w(c(1/5)) comp.weight	1.00			0.25			0.03		

Table F.9: Culture factors and capability / compatibility values

The required and available skills set and knowledge are defined (see Table F.10). The PM A shows more capability $v(s)$ = 64% than (PM B) is $v(s)$ = 68%, but less compatibility with available weight when knowledge experience is removed and only skills experience is retained (PM A $v(s)$ = 36%). For example, a project manager's profile can be defined in term of experience in a programming languages (Pl) as

$$Pl = \{(e_i, < \varepsilon_i, v_i, \omega_i >), i = \overline{1,4}\}$$

e_i : Programming languages (Oracle Tools)

ε : Existence experience (4 years)

v : Skills or scale levels (senior = 2)

ω: Weight is approximately equal (0.25)

i : Case number for i th factor (case no. 1)

PM A = {“programming languages (Oracle tools)”, <4, 2, 0.25>}

Factor	Required			PM A (average Cases)			PM B		
Skills	attrib	prop_level	weight =1/4	attrib	prop_level	weight =1/4	attrib	prop_level	weight =1/4
Programming Languages	5	3	0.25	4	2	0.25	4	3	0.25
Databases	3	3	0.25	3	3	0.25	2	3	0.25
O.S.	5	3	0.25	5	3	0.25	3	2	0.25
Experience	10	3	0.25	5	2	0.25	10	3	0.25
others	-	-	-	-	-	-	-	-	-
Skills Capability	v(s)	w(s)		v(s)	w(s)		v(s)	w(s)	
v(s1)	0.25	1.00		0.09	0.36		0.20	0.80	
v(s2)	0.25	1.00		0.25	1.00		0.17	0.67	
v(s3)	0.25	1.00		0.25	1.00		0.07	0.27	
v(s4)	0.25	1.00		0.06	0.22		0.25	1.00	
Total v(s)	1.00			0.64			0.68		
Length	4.00			4.00			4.00		
p(s)-compatibility length	1.00			1.00			1.00		
w(s) - compatiblity weight	1.00			0.08			0.14		
w(s(1/3)) comp.weight available	1.00			0.36			0.14		

Programming Languages	Databases	Operating Systems	Experience	Levels
1 = "Microsoft tools"	1 = "MS Database"	1 = "Windows"	1-Junior	1-Low
2 = "Oracle tools"	2 = "Oracle DB"	2 = "Unix"	5-10-Senior	2-Medium
3 = "MicroSystems Sun tools"	3 = "IBM Database"	3 = "Linux Redhat"	>=10-Principal	3-High
4 = "IBM tools"		4 = "Sun Solaris"		
5 = "Other tools"		5 = "QNX, Linux, Win"		

Table F.10: The skills set accumulator year of experience and capability / compatibility values

It is noted that the integrated quality characteristics for capability and compatibility provide a definition of how important leader capability needs to be to fit the project, even though both leadership and culture seem important to project success. Table F.11 shows the importance of both profiles. These are based on examples with a small number of attributes and from real life situations. This approach allows the selection of the most suitable people for given projects in a formal way rather than by a random selection.

Profile	Total Capabilty (vi)	compatiblity weight (w)	Compabitblity w(1/available)
Leader	0.77	0.21	0.27
Culture	0.70	0.07	0.25

Table F.11: Compatibility and capability for leader and culture weight

Appendix G

Papers Published

An Investigation into the Gulf States Government Approaches to Software Development and Effort Estimation, K. Hamdan, F. Abu Sitta, J. Moses, P. Smith, BCS,SQM 05.

A Software Cost Ontology System for Assisting Estimation of Software Project Effort for Use with Case Based Reasoning, K. Hamdan, H. El Khatib, J. Moses, P. Smith; IEEE Innovations in Information Technology, Dubai Nov 06.

BCS SQM 2005

An Investigation into the Gulf States Government Approaches to Software Development and Effort Estimation

Khaled Hamdan[1], Fawaz Abu Sitta[2], John Moses[3], Peter Smith[4]

[1] Information Technology-UGRU
P. O. Box 17172, United Arab Emirates University, UAE
e-mail: khamdan@uaeu.ac.ae

[2]Information Technology Division, Abu Dhabi Tourism Authority
P.O. Box 59333 Al Ain,United Arab Emirates
e-mail: fawazhazza@alain.ae

[3]School of Computing and Technology, University of Sunderland
St Peter's campus, Sunderland SR6 0DD, United Kingdom
e-mail: john.moses@sund.ac.uk

[4]School of Computing and Technology, University of Sunderland
St Peter's campus, Sunderland SR6 0DD, United Kingdom
e-mail: peter.smith@sunderland.ac.uk

Abstract

Billions of pounds are spent annually in an effort to e-enable governments' public services. The approaches that those governments follow in managing software cost estimation differ and are affected by many factors, such as the cultural environment, political system, leadership and economic strength. Since the Arabian Gulf States have similar factors to those mentioned above, the researchers have conducted their investigation in the capital city of UAE and believe that the results are alike in all Gulf States, except Dubai, as the character of its factors is different. In this study, 70% of the government departments in Abu Dhabi have been investigated. It was found that software development projects were over-budget, passed the deadlines or were lacking some of the required functionality. The absence of a well-defined model and the lack of software cost estimation skills, as well as the inflexibility of rules and regulations are the main reasons for these failures.

1.0 Introduction

Governmental departments worldwide spend billions of pounds annually in building information systems or in e-enabling their services. The approaches that those governmental departments follow in managing the software cost estimation differ and depend on many characteristics. Those characteristics are directly affected by the country's political system, economic strength, social behavior and culture and leadership.

In an effort to determine the most applicable model for software cost estimation in governmental departments, an understanding of the current approaches used in managing software development is necessary.

Arabian Gulf States (Bahrain, Oman, Saudi Arabia, Kuwait, Qatar and United Arab Emirates) have similar political systems, economy and social culture. Consequently, the researchers have conducted their study in Abu Dhabi and they believe that the results produced by this study are applicable to all the Gulf States.

In this paper sample departments have been selected from Abu Dhabi, the capital city of United Arab Emirates. These represent 70% of the government departments in Abu Dhabi and they share 73% of the IT spending in Abu Dhabi. Those departments have been studied very closely for a period of four months. A comprehensive survey was created and sent to those departments. The survey has been completed and validated through one-to-one meetings and site visits.

2.0 Background

Information technology plays an important role in the services sector of the United Arab Emirates (UAE) economy. In recent years the services sector has little to show for its spending on technology, however, the role of technology spending will be increasing in the coming few years [1,2,3].

According to a United Nations (UN) report, the UAE ranks among the leading countries in IT development and is ahead of many countries [1,2]. The UN declared in 2002 that the UAE was top in the Arab countries and ranked the 21st worldwide in measuring levels of e-Government. The Emirates continued to lead the Arab world in the year 2003, reaching the fourth generation of transactional, interactive e-Governments, which is only a step from the integrated version of e-Government. The achievement is mainly attributed to the politically driven implementation infrastructure of the telecommunication sector. UAE has also attained an outstanding level in standards of human development and distribution of population in the country. The Emirates provides comprehensive educational programs, which cover all segments of society [2].

The UAE has an open economy with a high per capita income and a sizable annual trade surplus. Its wealth is based on oil and gas output. The IT spending for the year 2003 was about 12.2 million pounds sterling on IT hardware, and expenditure for UAE was estimated at about 13 billion pounds sterling, including capital expenditures of 1.8 billion pounds sterling [4,5].

The United Arab Emirates will maintain its pan-Arab leadership in overall information and communication technology (ICT) use and development in 2008, according to the findings a study conducted by Madar Research [8]. The study described below also estimates the UAE IT market size in 2003, which is dominated by Abu Dhabi, and also forecasts market growth for 2008.

According to the Madar study, research and analyses indicated that Abu Dhabi represents 40% of the UAE total Information Technology labor market [9]. UAE Information Technology is an emerging market and has yet to reach the saturation stage. The pace of Information Technology development has been firmly set by the political will which is the core force driving the migration of both government and businesses towards the digital economy. The UAE Information Technology market was estimated at 753 million pounds sterling at the end of 2003. Abu Dhabi emerged as the biggest Information Technology spender in UAE, with total IT spending in 2003 valued at 344 million pounds sterling or 46% of the Information Technology market. Abu Dhabi spent about 35% of its total Information Technology expenditure on Information Technology services, valued at 121 million pounds sterling, 20% on data communications valued at 69 million pounds sterling, 28% on computer equipment valued at 95 million pounds sterling, and 17% on software products valued at 59 million pounds sterling in 2003[8].

Overall, UAE per capita Information Technology spending in 2003 is estimated at 188 pounds sterling per capita, much higher than the world average, estimated at 109 pounds sterling. In terms of the Cooperation Council for the Arab States the Gulf (GCC) (average per capita Information Technology spending roughly £100), the UAE's per capita Information Technology spending ranked second after Bahrain [8].

In the Global Information Technology Report 2002-2003[10], researchers for the world Economic Forum, gave the UAE a full score on the role of its leaders in creating an environment that is conducive to development of ICT. The UAE was the only Gulf Cooperation Council country receiving such a score, attributable to availability of a clearly spelled out ICT strategy, operational ICT-dedicated research facility, technology incubator, and Technopole initiative (*Technopoles: refer to geographically defined entities aimed at fostering technology and expertise transfer such as technology parks, innovation centers and high – tech clusters*) [10].

The UAE values today the highest level of ICT use in the Arab World. Madar Researchers give the Emirates the highest score among Arab countries, with mobile phone penetration standing at a high 68% and Internet at 30% - three times the world average of 10% [10].

3.0 The Approach

In this study, 70% of the Government departments in Abu Dhabi have been visited. The software development spending in those departments represents the 73% of the total spending in software development in Abu Dhabi.

A comprehensive survey has been prepared in an effort to determine the strategies that government departments have implemented for managing software development effort estimation.

The survey focused on collecting related information to type of software systems that have been developed in the government departments, as well as to variances between the actual cost and estimated cost. In this survey the researchers have been trying to identify the key reasons behind the variance between the actual cost and the estimated ones, see questionnaire in *Appendix 1.*.

In an effort to ease the process of collecting the information the surveys have been designed in a multiple-choice format and have been completed in one-to-one meetings. In order to validate the accuracy of the collected information site visits were arranged. The information has been collected and validated in a three-month period. The researches have spent this time in validating and auditing the accuracy of the collected information.

The Figure 1 shows the responses that have been received from the departments.

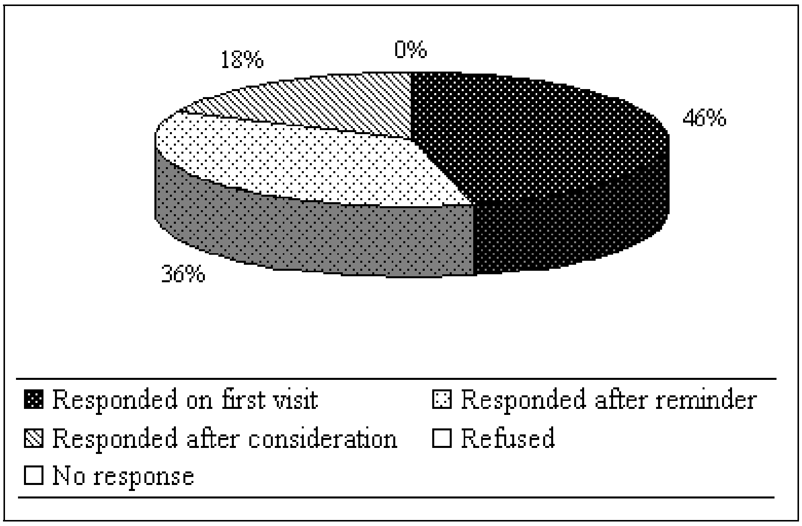


Figure 1: The overall response rates from Abu Dhabi Government Departments

4.0 The Data

In order to determine the various approaches that government departments follow in estimating the software development efforts, the researchers have started their study by inquiring about the approaches that the departments have chosen in developing their applications. The researchers have also looked at the various types of the applications developed in those departments. They have also investigated where most of these efforts have been expended to minor or major upgrades systems, or in developing new systems, see *Appendix* 1.

In this research the relation between the government departments and the software development vendors has been studied carefully. In an effort to understand the weaknesses and the strengths of the software development cost estimation models adopted in the government departments, the researchers have spent a considerable amount of time collecting data related to how the cost estimation process was conducted, in which phase it has been carried out and what the skills were that had been utilized to achieve the process.

In addition to the above, the researchers have investigated the main reasons behind the discrepancy between the estimated efforts and the actual ones.

5.0 Analysis

Despite the fact that most of the government departments' core business are not related to software development, it has been observed that more than half of those departments have used enormous resources to develop their software systems in-house and with labor of their own technical employees. The other option, outsourcing systems software development, has not proved to be very popular among most of the government departments. A very small number of government departments have chosen both options and developed in-house the software systems that consist of the most critical information, whereas non-core applications are outsourced.

Three main reasons have culminated in government departments' preferring the utilization of their own resources and developing their systems in-house: the first reason stems from the lack of a national law governing software developments and digital signatures: the second because of the lack in the specialized skills in developing strong Service Level Agreements (SLAs): thirdly, opportunities for finding outsourcing software development vendors are limited. Figure 2 shows the proportions of each option adopted in the Abu Dhabi.

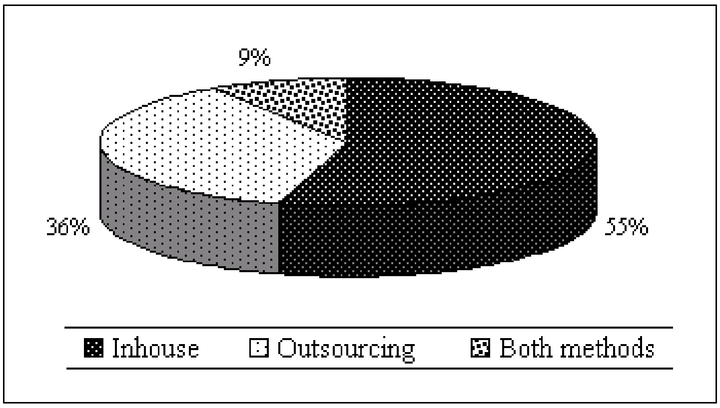


Figure 2: The proportion of applications developed in-house, outsourced and Both Methods

Most of the government departments surveyed started automating their internal systems in the mid 80's; consequently, enterprise resource planning (ERP) systems development has come to occupy the largest portion of software development and utilization applications in government departments. E-enabling the services that are provided to the public is still in the early stages of development and has not been taken into account in the survey.

Not withstanding this, it is expected that in the next few years most of the government departments will pay considerable attention to providing their service through electronic means. Figure 3 presents the various types of software systems that are developed in government departments.

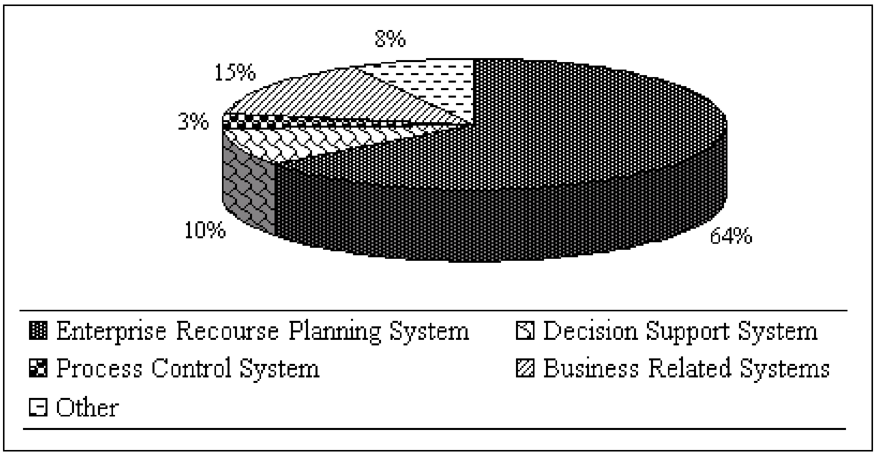


Figure 3: The proportions of applications types developed for the government departments

It has been noted that more than 50% of the departments interviewed have replaced old systems with new ones, rather than upgrading them. It is crucial to note that benchmarking with historical data has not been considered in depth when developing new systems. Figure 4 shows the different types of development activities that have been carried out in system developments.

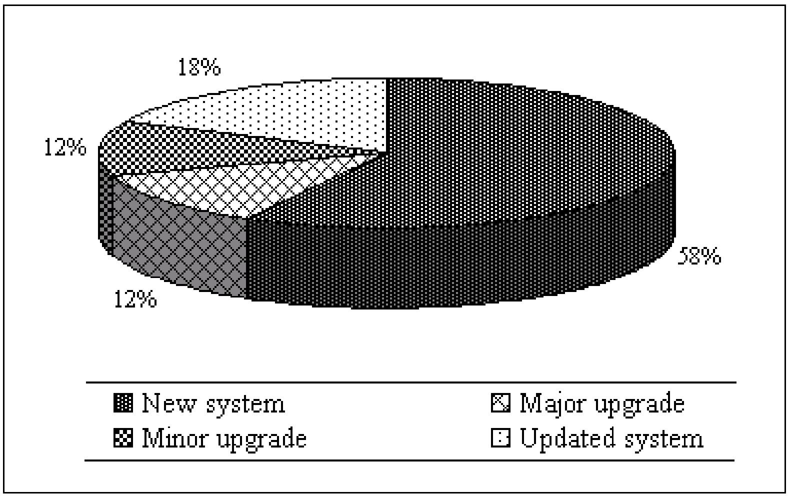


Figure 4: The anticipated delivery of projects

Even though the majority of the government departments surveyed have developed their own applications in-house, only a small number of those departments have paid major attention to software development effort estimation. Departments have been forced to allocate their IT budget at the beginning of each fiscal year without focusing on the software development effort estimation, this being due to the rules and regulations that control the budgeting process. As shown in Figure 5 most departments calculate the cost of the production environment rather than focus upon the development effort.

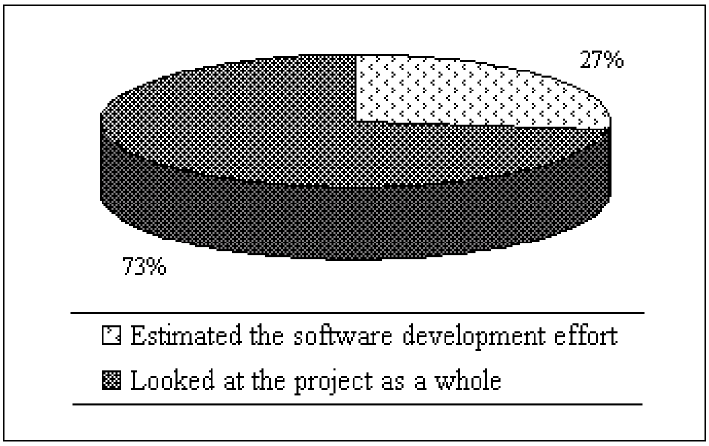


Figure 5: Departments that paid attention to software development effort estimation vs. departments that didn't pay major concerns to software development effort estimation

It has been observed that 73% of the departments had signed time-based contracts with vendors rather than deliverables-based ones when outsourcing software development, see Figure 6. None of those vendors or the departments have estimated the software development effort or based their assumptions on a solid and recognized estimate model. Consequently, all of those projects without exception have failed and vendors have not been able to deliver the application on time, within the allocated budget or fell short of some of the required functionality.

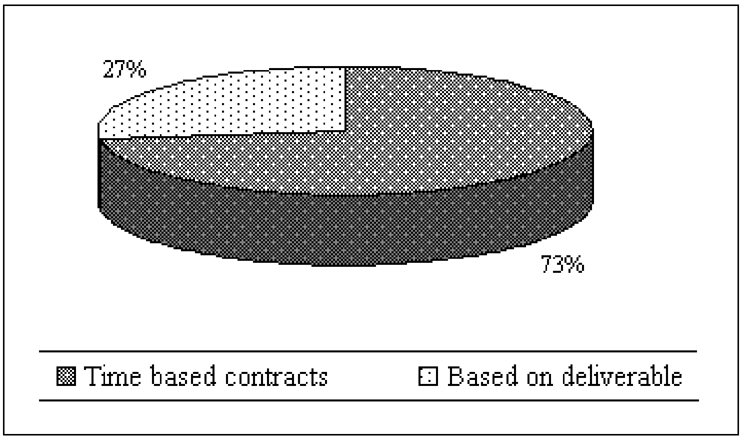


Figure 6: Tme-based contract vs. deliverable ones

Large numbers of departments have stated that they have a major discrepancy between the estimated and the actual cost for their software systems development. This weakness can be attributed to the fact that departments lack expertise in the field of software effort estimation. The preferred approaches that are followed by project managers inflate the project budget. Figure 7 shows the discrepancy between the estimated effort and the actual ones.

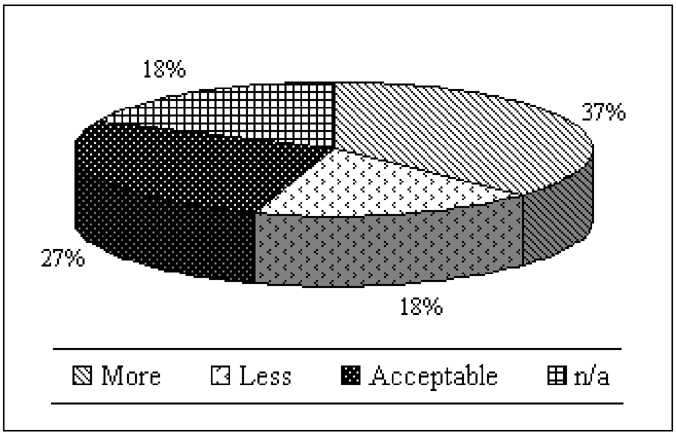


Figure 7 Discrepancy between the estimated effort and the actual ones
Acceptable: (Variance between actual effort and estimated one are negligible)

The chart below (Figure 8) further demonstrates the reasons behind the discrepancy between the actual and estimated effort. It is obvious that unanticipated tasks account for the main reason for the discrepancy. The results demonstrated here reflect the urgent needs for proper estimation model that should be followed when developing software systems in government departments.

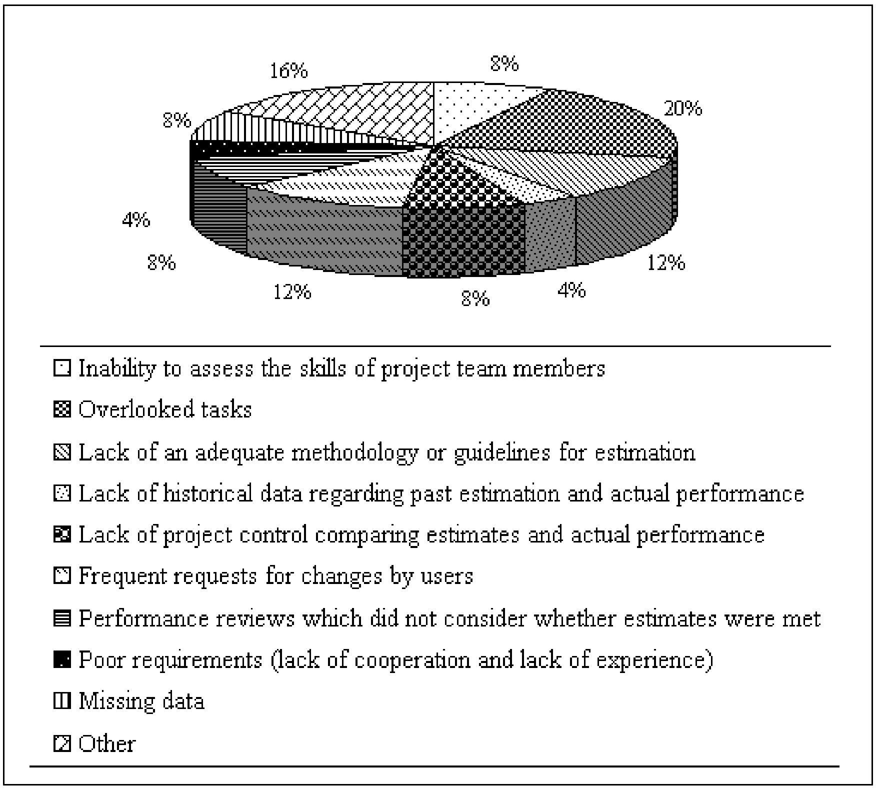


Figure 8: Reasons behind the effort estimation discrepancy

The other reasons for the discrepancy between the actual effort and the estimated one are due to the fact that software development estimation is conducted at the early stages of a project. Project managers are obligated by the rules and regulations that force them to set the software development budget prior to, or at the early stages of the project kickoff. Figure 9 below indicates that software development starts at the early stages of a project.

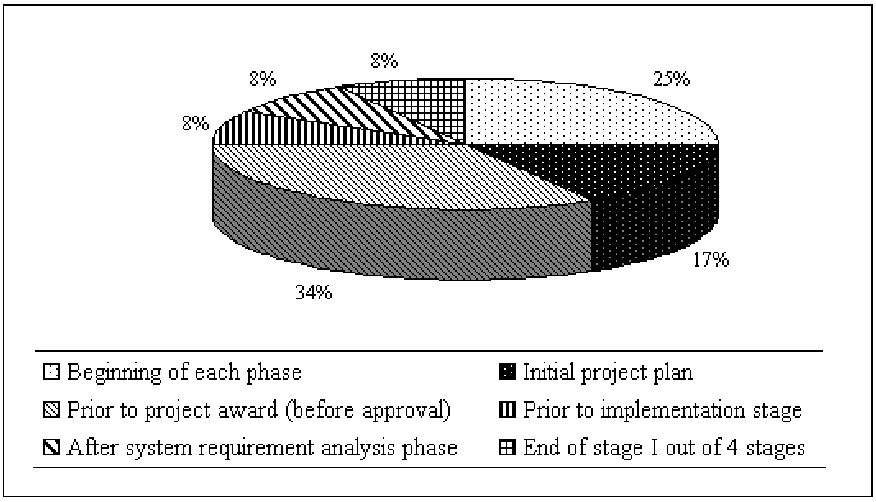


Figure: 9 When effort estimation occurs during a project

In order to get more precise estimation and to avoid any risks, project managers prefer to perform some informal cost estimation. However, they tend to reduce small elements of the project scope rather than request revision of their budget. Figure 10 shows when the re-estimation process occurred during the project life cycle development.

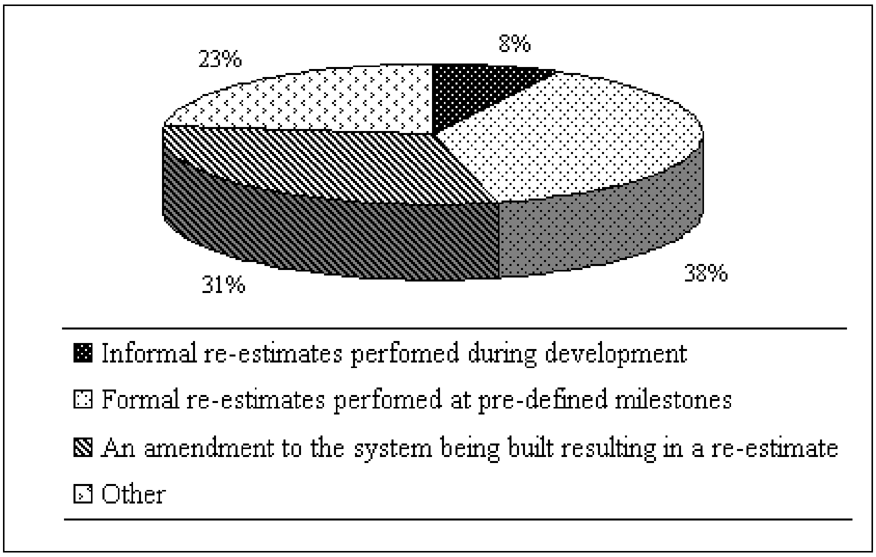


Figure 10: Reasons for cost re-estimation

Figure 11 shows the parameters that have been re-estimated.

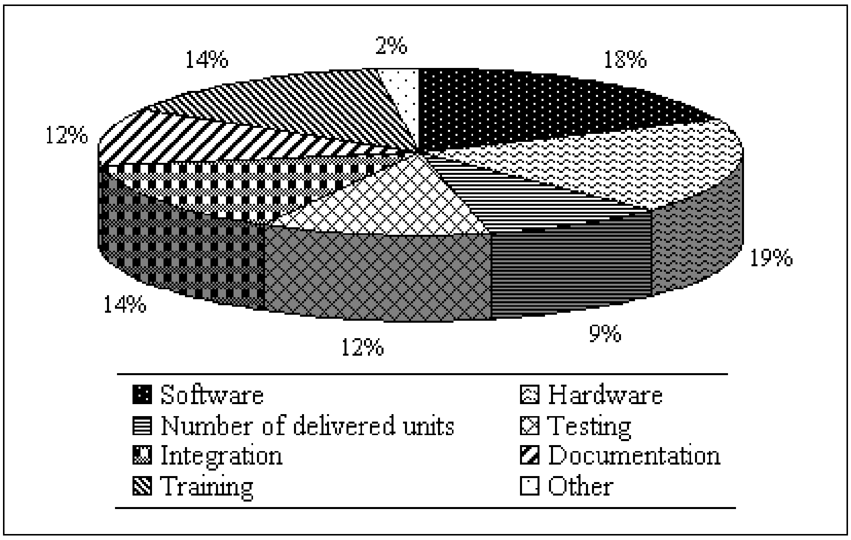


Figure 11: Estimation Parameters

Project managers and leaders beside a large proportion of their IT teams were involved in the informal estimation (as indicated in the Table 1). Despite this diverse allocation of personnel only 5% of them have previous experience in cost estimation. It was noted, too, that some software cost estimation models have never been used in any of the projects or the departments surveyed.

IT Position
Project manager (PM)
System manager & system analyst
Technical advisor
Support team
Project administrator
Head of applications
Team leaders application of software
IT consultant
Network engineering officer
Programmers & SW developers

Table 1: IT players in cost estimation

Also, it was observed that 40% of project managers took advanced training in project management, whereas, 60% present did not, Figure 12. However, none of them had undertaken a course on software cost estimation.

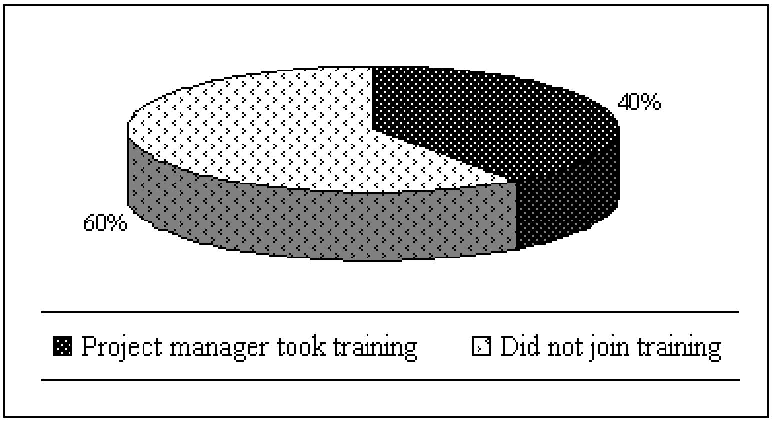


Figure 12 the proportion of project managers who attended Project Management training and those who did not attend any training courses

Finally, it was found that almost 90% of the teams involved in cost estimation tasks strongly recommended that they should participate in effort estimation training. This result reflects a major demand for offering specialized courses about software development cost estimation.

6.0 Discussion

Well defined national e-law and the availability of specialized skills in developing and managing Service Level Agreements (SLAs) are the key problems that drive the government departments not to outsource their software development. Solving those problems will help the departments to focus on their core business rather than spending a considerable amount of resource in managing the tiny details of software development.

After examining the data collected in this study the authors recommend the following in order to improve the effort estimation process.

Departments are requested to document their learned lessons at the end of each project. This will help to maintain more accurate software cost estimation for both the vendors and the departments.

Vendors and Departments should build and maintain a database to store all the historical data related to the completed Information Technology Projects. These sets of data should be used as a guideline for planning to new projects.

In order to be more accurate in maintaining software development effort estimation, a national database must be built and shared among all the government departments and vendors. It is believed that this database will provide very helpful and accurate estimates since the software development projects are similar in the government departments (please refer to the types of applications that are developed in the government departments Figure 3, as well as to the reasons behind the discrepancy between actual and the estimated ones mentioned in Figure 8).

In addition to the above, building this database will help in fulfilling the shortage in the cost estimation engineers that the government departments lack.

Awareness programs highlighting the importance of software development estimation should be conducted for senior management, project managers and staff. This will help in convincing the government department higher authority to invest in building and maintaining such database.

7.0 Conclusion

In this study, it was found that the software development projects were either over-budget missed the deadlines or fell short of some of their required functionality. The reasons behind these results may be summarized as follows:

1) Software cost estimation occurred at the early stages of the project and was consequently not based on a well-defined cost estimation model.

2) People in charge of software development lack the experience and the skills in software cost estimation.

3) Rules and Regulations that for e-Government and Information Systems are not defined well.

Clearly, in the UAE and elsewhere in the Gulf, leaders are keen and enthusiastic to seriously consider and implement more advanced models to software cost estimation.

The researchers concluded that by implementing advanced national database that stores all the historical data and lessons learned from the completed projects, will help greatly in maintaining more accurate estimation and in fulfilling the needs for specialized software cost estimation engineers. It is believed that this database would save the Gulf States a considerable amount of money.

8.0 References

1 United Nation Report-Benchmarking e-Government: Global Perspective, website, www.unpan.org/middleeast.asp
2 E4all, Dubai e-Government magazine, no 9, pp 1-17, July 2004
3 E4all, Dubai e-Government magazine, no 2, pp 2-3, December 2003
4 Abu Dhabi Chamber of Commerce & Industry, website, www.adcci-uae.com
5 CIA The world Fact book, website, www.cia.gov/cia/publications/factbook/
6 Jacqueline Webb, Social & Economic Developed in The Emirate of Abu Dhabi., p23-77, Economic Horizons, ol 25
7 Maddy Reddy, The Information & Technology Publishing Co. Ltd., IT security woes fuel software vendor sales, Arabian Computer News newspaper, October 2004, website, www.itp.net
8 Madar Research Journal, eLandscape, issue 10, December 2003, UAE IT, p25-34, website, www.madarresearch.com
9 IT Labor Market, Dubai Knowledge Economy 2003-2008 Vol II, p67-71
10 Macro-economy & IT Market, Overview & Dubai Competitiveness, Madar Research group, Dubai Knowledge Economy 2003-2008 Vol III, p7 –23
11 Torben Pedersen, UAE IT Services 2004-2008 Forecast and 2003 Vendor Shares, IDC Analyze the Future, 2003, website, www.idc-cema.com

Appendix 1

University of Sunderland

A Survey of Software Development Effort Estimation in

Abu Dhabi Governmental Departments

Prepared by:

Khaled Hamadan
for
his PhD thesis
Supervisors:

Prof. Peter Smith, University of Sunderland, UK
Dr. John Moses, University of Sunderland, UK

Dr. Fawaz Abu Sitta, Senior Consultant, Dubai e-Government

Introduction

Currently I am conducting a study to investigate the process used to estimate the effort of developing and maintaining systems involving significant amounts of software. The objectives of this study are to determine the current state-of-art in software effort estimation techniques to help identify methods and researches directions for improving software effort estimation and control.

I wish to look at software development in various locations in the UAE. My approach is to identify a set of development and maintenance projects, which have been completed (or are close to completion), to determine when and how the software costs were estimated, and then to look at how the estimated effort relates to the actual effort.

I am interested to look at and study all of the software systems projects that you have and will be developing.

This document contains a set of topics and questions that are intended to gather information about specific projects, which has been undertaken by your organization.

The following questions will help me to determine the effort spent on software development. You can be assured that all replies will be treated in the strictest confidence, and that no organization or individual will be named, without prior approval, in any reports produced as a result of this questionnaire. If you are interested in receiving a copy of the collated results from this study, please let me know during the interview.

To make further inquires please feel free to contact me at the following Address:

Khaled Hamdan
P O Box 17172
United Arab Emirates University, IT Unit-UGRU
Al Ain
United Arab Emirates
Khamdan@uaeu.ac.ae
Tel: 971(50) 6436350

Respondent Information

Company Name: ______________________________________

City / Emirate: ______________________________________

Contact Person (*responsible for filling out this questionnaire*)

Name: ______________________________________

Job Title: ______________________________________

Email: ______________________________________

Telephone: ______________________________________

Remarks: ______________________________________

1. Was the project developed totally in-house or was it contracted to an external vendor?

   ☐ In-house
   ☐ Out sourced

2. Does your department have developed or used any of the following systems?

		Yes	No
f)	Enterprise Recourse Planning System	☐	☐
g)	Business related Systems	☐	☐
h)	Control System	☐	☐
i)	Decision Support System	☐	☐
j)	Other (Please state) ___________________________		

3. What was expected to be delivered at the completion of the project?

   ☐ A new system was required
   ☐ A major upgrade: a currently existing system was upgraded, requiring extensive amounts of new software to be developed
   ☐ A minor upgrade: a currently existing system was upgraded required (some) software
   ☐ A system was updated by modify existing software

4. Which one of the following methods have you selected while allocating the budget for the IT projects?

   ☐ Estimated the software development effort separately and add it to the total cost of the project
   ☐ Didn't estimate the software development effort and instead looked at the project as a whole

5. What was the contracted effort?

   ☐ Time based contract
   ☐ Based on Deliverables

6. How close was the contracted effort to the bids? (Please state).

   ☐ More
   ☐ Less
   ☐ Acceptable (variance between actual effort and estimated one are negligible)
   ☐ Not applicable (N/A)

7. In your opinion, what were the reasons for the discrepancy between the actual effort and the estimates?

- ☐ Inability to anticipate skill of project team members
- ☐ Overlooked tasks
- ☐ Lack of an adequate methodology or guidelines for estimating
- ☐ Lack of historical data regarding past estimates and actual performance
- ☐ Lack of project control comparing estimates and actual performance
- ☐ Inability to tell where past estimates failed
- ☐ Frequent requests for changes by users
- ☐ Performance reviews don't consider whether estimates were met
- ☐ Missing Data
- ☐ Other (please state) ___________________________

8. At what stage of the project life cycle did this effort estimation occur?

- ☐ Beginning of each phase
- ☐ Initial project plan
- ☐ Prior to project award (before approval)
- ☐ Prior to implementation stage
- ☐ After system requirement analysis phase
- ☐ End of stage 1 out of 4 stages

9.1 Were any re-estimates of effort performed during development?

☐ Yes ☐ No

9.2 If Yes to 9.1, then please identify when the re-estimating procedures were invoked (see list below). If possible, indicate when (in the development life cycle) the re-estimates occurred, and the result of the re-estimates. (Please tick appropriate box)

- ☐ Informal re-estimates performed during development.
- ☐ Formal re-estimates performed at pre-defined milestones.
- ☐ An amendment changed the system being built and a re-estimate was required.
- ☐ Other (Please state)_____________________________________

9.3 Which items were included in the re- estimates? (Check all that apply)

- ☐ Software
- ☐ Hardware
- ☐ Number of delivered units
- ☐ Testing
- ☐ Integration
- ☐ Documentation
- ☐ Training
- ☐ Other (please state) ___________________________

10. List the job titles of the people whom involved in the estimation process.

11. Did the people involved in the estimation process have any previous experience in software effort estimation before they were called upon to estimate for this project?

    ☐ Yes ☐ No

12. Was any training in project management provided for individuals?

    ☐ Yes ☐ No

13. Is such training required in your opinion?

    ☐ Yes ☐ No

14. If training was provided, do you think the training improved your organization's ability to arrive at accurate effort estimates?

    ☐ Yes ☐ No

15. I would be grateful for any other information that you could offer, in order to clarify your responses, or to make further suggestions for the study. This may include questions that were not asked or were inappropriate to your department.

    ____

16. Would you be willing to take part in a further interview?

    ☐ Yes ☐ No

17. Would you like a copy of the results of the survey?

    ☐ Yes ☐ No

Thank you for taking part in this survey and for your co-operation

A Software Cost Ontology System for Assisting Estimation of Software Project Effort for Use with Case-Based Reasoning

Khaled Hamdan[1], Hazem El Khatib[2]
United Arab Emirates University
[1]*khamdan@uaeu.ac.ae*
[2]*Hazem@uaeu.ac.ae*

John Moses[3], Peter Smith[4]
University of Sunderland
[3] *john.moses@sund.ac.uk*
[4]*peter.smith@sunderland.ac.uk*

Abstract

Software project cost and effort estimation has become an increasingly important field in the past years due to the overwhelming role of software in today's global market. Several studies have been dedicated to create models in order to estimate the effort of software development. Most of the studies focused on expert judgment, analogy, parametric and algorithmic methods, bottom-up methods, and top-down methods. Nearly all estimating methods need information about how projects have been implemented in the past. However, this information may be of limited use to estimators, as there are uncertainties in the way that various terms, variables and factors are being interpreted. Two projects that may seem similar may indeed be different in a critical way. Moreover, the uncertainty in assessing similarities and differences means that two different analysts could develop significantly different views and effort estimates. The major contributions this paper makes are: 1) identification of an ontology-based cost estimation process framework for defining the semantics of project development data; 2) introduce the culture factor as it affects the software effort estimation; 3) development of a Software Effort Estimation Ontology System (SEEOS) for use in estimating software project cost in a group of organizations. The system establishes a set of common project parameters between different projects and provides a common understanding of project parameters and their semantics. This system enables project managers to elicit software project features that are semantically compatible with new project requirements. The system has been implemented using Java and a relational database management system and data which have been collected from within UAE companies using an online system.

Index Terms: Project effort estimation, Ontology engineering, Case-based reasoning.

1. Introduction

Software project cost and effort estimation is an increasingly important field due to the overwhelming role of software in today's global market[1]. However, there is no optimal approach to accurately predict the effort needed for developing a software system[2]. Usually, the information gathered at the early stages of software system development is insufficient for providing a precise effort prediction. Even if data exist for a software system that appears analogous to the new case there will be new facets of development in the new case. For example, the system is likely to use different developers and internal company politics surrounding development may be very different. This means that it may be practically impossible to produce an accurate estimate of system development effort at an early stage in the project. Yet other problems which can affect accurate estimation include[3].

- Novel application of the software
- Changing technology
- Lack of homogeneity of project experience
- Subjective nature of estimating
- Political implication

In this study we identify a difficulty, which to the authors best knowledge has not been widely considered as a factor affecting estimate accuracy “Cultural implications.” We use the Gulf States as an example of a group of countries with a particular cultural type, and this culture may have had an impact on the accuracy of software effort and cost estimation. The Gulf States all have similar cultural environments, political systems, leadership and economic strengths. Within the Gulf States, the United Arab Emirates (UAE) has been considered as one of the fastest growing countries in utilizing, adopting and developing Information Technology in the Gulf States and the Middle East. Local and Federal government agency has spent millions of dirhams on internal software development. However, a big portion of those developed applications either went over budget or have failed to be delivered on time[4].

Our research to date has identified that there is no single formal estimation method or model in use for software cost estimation[4]. From questionnaires completed by the various organisations in the UAE, it appears that failures and over-runs in most of the application development projects have been attributed, by the questionnaire respondents, to lack of an estimation model. We therefore recommend that a common software cost estimation model, which takes into account the culture of the countries, should be constructed for the UAE and the Gulf States. The first step towards creating this model has been determined by identifying the parameters of the model.

This study has identified the importance of organisational and cultural factors and project leadership for improving effort estimates by Analogy. In the UAE, senior management or functional managers (department managers) have more authority than the project managers, and this influences the process of cost estimation. We intend to use our newly identified factors, such as project management authority and privileges, to increase the range of factors considered and the scope of usage of Case-Based Reasoning methods and in so doing to improve the accuracy of existing effort estimates in the Gulf States. For example, Shepperd’s Case-Based Reasoning (CBR) model is seen to be a model that could be suitably adapted for use in the UAE[2].

A large number of researchers[1,2,5] and others have tried to develop models for measuring and estimating software development effort. Few of those models have considered or focused on the cultural issues within the organization or the leadership characteristics of the project managers. This research proposes the hypothesis that organizational culture and project leadership is a very significant factor in determining the precise effort required for software development projects. This is particularly true within the Arabian Gulf States. In order to determine and validate the hypothesis, a survey of software development projects within government departments in the Gulf States has been undertaken, and the impact of organizational culture and leadership on software effort estimation has been determined.

2. Approaches to Cost Estimation

There are various techniques used in software cost estimation. Examples include the Delphi method[6] and Nelson's SDC[7]. In the 1970’s and the 1980’s, more models such as TRW Wolverton[8], SLIM[9], Checkpoint[10], PRICE-S[11], SEER[12], ESTIMACS[13] and COCOMO[14] were developed. All these models faced the same problem where software increased in size and use, it also expanded in complexity, making it very difficult to accurately predict the cost and the quality of software product. Barry Boehm[15] identified the main ways of deriving estimates of software development effort as:

- Algorithmic models, which use effort drivers representing characteristics of the target system and the implementation environment to predict effort.
- Expert judgment, where the advice of knowledgeable staff is solicited[16].
- Parkinson, which identifies the staff effort available to do a project and uses that as the estimate.
- Price to win, where the estimate is a figure that appears to be sufficiently low to win a contract.
- Top-down, where an overall estimate is formulated for the whole project which is then broken down into the effort required for component tasks.
- Bottom-up, where component tasks are identified and sized and these individual estimates are aggregated.
- Analogy, also CBR, where a similar, completed, project (source cases) is identified and its actual effort is used as the basic of the estimate for the new project (target case).

Here the Analogy model is considered. Methods using analogy rely on data from actual projects, avoiding expert judgment's reliance on recall. The authors[2] also avoid the complexity of parametric/algorithmic models. They intended to provide some experimental data to assist with the more effective use of CBR techniques for building prediction systems[2]. Through their study, they concluded that the overall picture of their research suggested that estimation by analogy tends to be a more accurate prediction method. They also concluded that collecting historical data is sufficiently challenging and they prefer to allow estimators the freedom to utilize those features that they believe best characterize their projects and that is most appropriate to their environments using Euclidean distance as a means of measuring similarities between cases[5].

In this research, the similarities and differences between the different projects' features and the source case that is nearest the target by measuring the distance between cases adapted from[2] as shown in equation 1 and equation 2 was identified.

$$Sim(C_1,C_2,F) = \frac{1}{\sqrt{\sum_{j \in P} Feature\_Dissimilarity(C_{1j},C_{2j})}} \quad (1)$$

F = Set of features

C_1, C_2 = Cases

(2)

$$Feature\_Dissimilarity(C_{1j},C_{2j}) \begin{cases} (C_{1j}-C_{2j})^2 & \text{Features are numeric} \\ 0 & \text{Features are categorical and } C_{1j}=C_{2j} \\ 1 & \text{Features are categorical and } C_{1j}\neq C_{2j} \end{cases}$$

Tools, such as Bournemouth University's ANGEL[2], can be used to support the estimation. However, there are several disadvantages to using such analogies some of which are:

- Similar projects may not exist.
- Historical data may not be accurate.
- Case data can be hard to gather.
- It is necessary to build a library of cases before the system can be useful.
- Cases require interpretation; two projects that may seem similar may indeed be different in a critical way. Moreover, the uncertainty in assessing similarities and differences means that two different analysts may have significantly different views and eventual estimates.

We have developed a Software Effort Estimation Ontology System (SEEOS) for helping to estimate software project cost estimation using ontology as shown in Fig 1. The left side panel shows the project's entities along with their attributes, descriptions and values. While the right side panel shows the selected entities by the project manager(s) to be estimated. This system was implemented using Java and relational database management system and the data, which were collected within UAE companies using an online system that has been developed as part of this project as show in Fig 2.

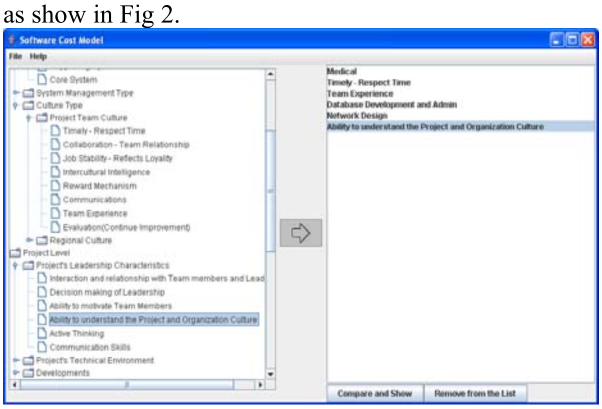


Figure 1: SEEOS User Interface

Figure 2: Database Online System

3. Software Cost Ontology

Nearly all estimating methods need information about how projects have been implemented in the past. There have been some attempts to set up industry-wide database of past projects, e.g. [17]. However, this data seems, to us, to be limited in use to estimators who are able to interpret and account for the uncertainties in the way that various terms and factors can be interpreted in the context of their own organizations. For example, the use of the term 'testing' needs to be carefully examined in the database and assessed within the context of the estimators system development practices and project constraints. Does it cover the activities of the software developer when debugging the code? Furthermore, does 'design' include drawing up a program structure diagram or does this come under the heading of 'software development'? [3] Care needs to be taken in judging the applicability of data to the project under consideration. Such judgments also need to be applied to possible differences in other environmental factors such as the methods of programming in the same language, how development standards are enforced, how to classify the experience of the staff, etc. The project managers must be able to select similar existing software projects based on well understood similarity features.

In this study we have developed an 'ontology of software development', which allows the semantics of software development to be classified by using our ontology of project factors. We believe the ontology will enable project managers to elicit a software project's features correctly. The semantics of a software project's development factors are elicited based on our ontology system, which guides the application manager's activities by using a set of definitions for each attribute included in the ontology system. Ontology defines the semantics of the project parameters and their relationships; it establishes shared conceptions between different projects. Fig 3 shows part of the developed ontology.

The ontology associates each Project type with an Organization's Line of Business as Sub Class Of, each SEEOS mode is associated with a set of entities and attributes associated with Type and values. The following is a description of the concepts used in the ontology:

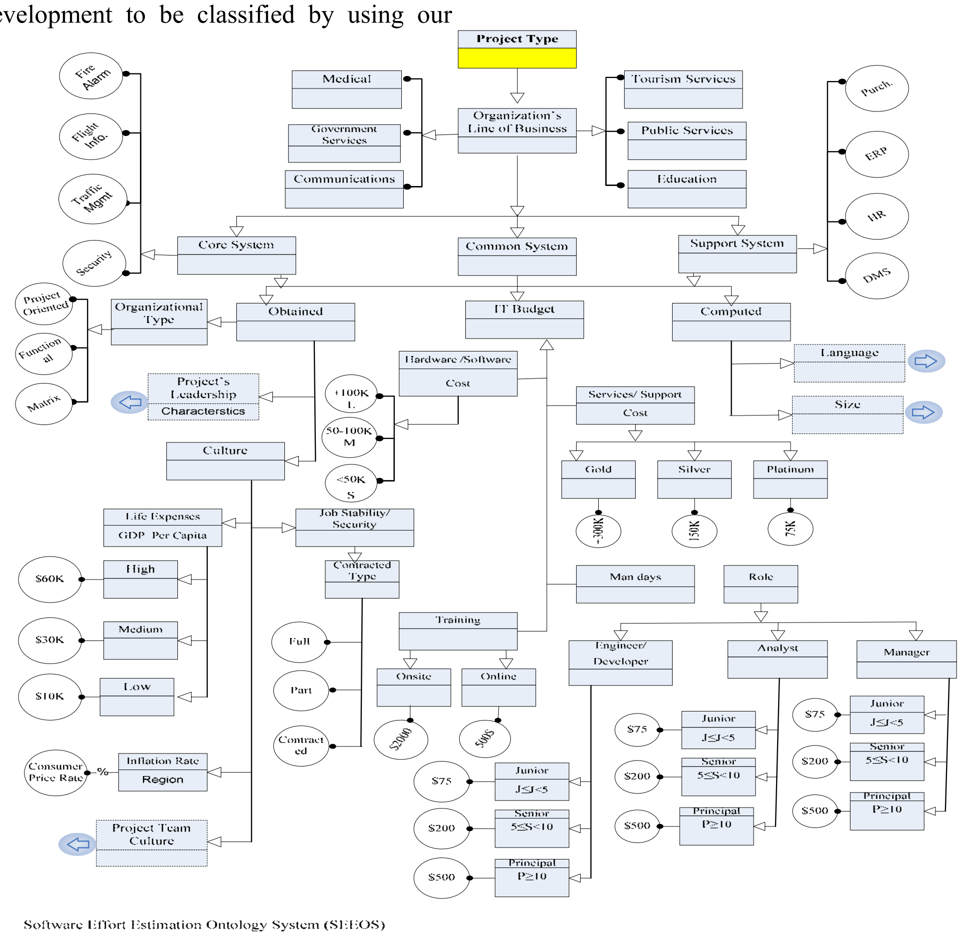


Fig3: Project cost estimation ontology

The system is categorized into Support, common, and Core Systems. Those systems have IT budget that has services, Hardware or Software Cost, mandays, and training, while training has online and onsite. Computed has Languages and Project Size. And obtained mode has Organizational Type, Project Leadership and Culture. Culture has Job Stability, Life expenses, Inflation rates, and Project team Culture.
The ontology classifies a junior developer as the one who has more than 1 year of experience and less than 5 years. Senior developer was identified as the one who has more than 5 years of experience and less than 10 years. A principal developer is someone who has more than 10 years of experience. Culture and leadership amongst other concepts have also been introduced as depicted in Table 2. The values of the corresponding Organizational Culture variables have been specified as follows: 1-3 (Low), 4-6 (Nominal), and 7-9 (High). For example, Timely means respecting time and the individual understands general views of time (event or relationship). If it is 95% of time, then the scale is 9. Low means that the individual slacks, or has frequent absences. Nominal is when the individual frequently comes late to work or for a meeting. The other attributes like collaboration, job stability, intercultural intelligence, reward, communications, and team experience have the same values on the scale.

Table 2: Organizational Culture

Organizational Culture	Low	Nominal	High
Timely	1-3	4-6	7-9

4. Conclusion

This paper proposes to reduce the risk of misunderstandings by unifying the terminology of the different stakeholders with the help of an ontology. In this study a SEEOS has been presented for assisting in the estimation of software project effort and cost. Software to support and enable the use of the SEEOS has been developed, which incorporates specific cultural terms, factors and issues that have been identified as an influence on cost estimation. The identified factor values will then be used in a CBR system for effort estimation. This system is expected to enable project managers to elicit software project factors, features and terms that are semantically equivalent to those used in a new project and for which effort and cost estimate are required. The efficacy of this SEEOS and its impact on the accuracy of effort estimates will be tested within the Gulf States and reported over the forthcoming years of its application.

5. References

[1] B. Kitchenham, , S.L. Pfleeger, N. Fenton, “Towards a Framework for Software Validation”, *IEEE Trans.* On SE, Vol. 21 No. 12, Dec 1995.

[2] M. Shepperd, C. Schofield, “Estimating Software Project Effort Using Analogies” I*EEE* Vol. 23, No, 12 Nov. 1997.

[3] B. Hughes, M. Cotterell, Software Project Management, *McGraw-Hill*, 2006.

[4] K. .Hamdan, , F. Abu Sitta,., J. Moses, P. Smith, “An Investigation into the Gulf States Government Approaches to Software Development and Effort Estimation”, *BCS-SQM* 2005.

[5] Angelis, Stamelos,“A Simulation Tool for Efficient Analogy Based Cost Estimation”, *Empirical Software Engineering*, 5, 35-68, 2000.

[6] Helmer-Heidelberg, “Social Technology”, Basic Books, New York, 1966.

[7] Nelson, E. A., “Management Handbook for the Estimation of Computer Programming Costs”, Systems Development Corp., Oct 1966.

[8] Wolverton, R.W, The Cost of Developing Large-Scale Software. IEEE Transactions on Computer, Volume C-23, No. 6, pp. 615-636, June 1974.

[9] H. Lawrence, Putnam and Ware Myers, “Measures for Excellence”, Yourdon Press Computing Series, 1992.

[10] C. Jones, “Applied Software Measurement, Assuring Product & Quality”, McGraw Hill, pp. 204-331, 1997.

[11] R. Park., “The Central Questions of the PRICE Software Cost Model,” 4th COCOMO Users’ Group Meeting, November 1988.

[12] R. W Jensen, “A Comparison of the Jensen and COCOMO Schedule and Cost Estimation Models”, Proceedings of the International Society of Parametric Analysts, pp. 96-106, April 1983.

[13] Rubin, “Macro-estimation of Software Development Parameters: the Estimacs System”, Techniques and Alternatives, IEEE Press, New York, pp.4-16, July 1983.

[14] B.W. Boehm, Software Engineering Economics, Prentice-Hall, 1981.

[15] B.W. Boehm, 1995, “Cost Models for Future Software Life Cycle Processes: COCOMO II,” Netherlands, Vol. 1, pp. 45 - 60, 1995.

[16] R. Futrell, D. Shafer, L. Shafer "Quality Software Project Management", Software Quality Institute, January 2002.

[17] ISBSG, 2003, International Software Benchmarking Standards Group, Data Repository, Release 8, site: http://www.isbg.org.

Appendix H: Glossary

List of Acronyms and Abbreviation

AEM	Algorithmic Estimation Models
AFP	Adjusted Function Points
ANOVA	Analysis of Variance
CBR	Case-Based Reasoning
COCOMO	Constructive Cost Model
DARPA	Defense Advanced Research Projects Agency
DBMS	Database Management Systems
DFD	Data Flow Diagrams
DoD	Department of Defense
E	Effort
EbA	Estimation By Analogy
EI	External Inputs
EIF	External Interfaces Files
EJM	Expert Judgment Models
EO	External Outputs
EQ	External Inquiries
ERD	Entity Relationship Diagrams
ERP	Enterprise Resource Planning
FPs	Function Points
GCC	Gulf Co-Operation Council
GDP	Gross Domestic Product
GL	Generation Language (2nd, 3rd, 4th)
GSC	General System Characteristic
HR	Human Resources
IEEE	The Institute of Electrical and Electronics Engineers, Inc.
IFPUG	International Function Points User's Group
IFPUG	International Function Point User Group
ILF	Internal Logical Files
ISPA	International Society of Parametric Analysts
IT	Information Technology
JSP	Java Server Page
LN	Natural Logarithmic
LOC	Lines of Code
LS	Leadership
MRE	Magnitude of Relative Error
MER	Magnitude of Relative Error to the Estimate
MdMRE	Median of the Magnitude of Relative Error
MMRE	Mean Magnitude of Relative Error
MMER	Mean Magnitude of Error Relative to the Estimate
N	New
NASA	National Aeronautical And Space Administration
O	Old
OR	Ordinal Regression
Org	Organisation
PCA	Principal Component Analysis
Pred(25)	Percentage Predicted with a MER less or equal than 0.25.
PRICE S	Price Software Model
QSM	Quantitative Software Management
R	Coefficient of Correlation

R2	Coefficient of Determination
S	Size
SE	Standard Error
SEEOS	Software Effort Estimation Ontology System
SEER-SEM	System Evaluation and Estimation of Resources
SLIM	Software Life-Cycle Model
SLOC	Source Lines of Code
SPR	Software Productivity Research
UAE	United Arab Emirates
UFP	Unadjusted Function Points
UFP	Unadjusted Function Points
UML	Unified Modeling Language
UN	United Nation
VAF	Value Adjust Function Points
WBS	Work Breakdown Structure