1998

Parametric software project estimating: An analysis of current practice

Stuart Hope
Edith Cowan University

USE OF THESIS

The Use of Thesis statement is not included in this version of the thesis.

Parametric Software Project Estimating - An Analysis of Current Practice

Master's Thesis

Submitted in Partial Fulfilment of the Requirements for the Award o
Master of Science (Computer Science)
at the School of Computer and Information Science
Faculty of Communication, Health & Science
Edith Cowan University

Student: Stuart Hope BSc (Hons) 0854327

Supervisor: J.E. Terry

October 1998

Declaration

"I certify that this thesis does not incorporate, without acknowledgment, any material previously submitted for a degree or diploma in any institution of higher education and that, to the best of my knowledge and belief, it does not contain any material previously published or written by another person except where due reference is made in the text".

Stuart Hope
October 1998

Acknowledgment

I would like to thank my supervisor, Julian Terry, for his support and interest throughout this work. I would also like to thank him for the meticulous and timely checking of the draft and final documents.

Abstract

As society and the world economy moves into the second millennium, the service industries involving knowledge workers will continue to increase. Software is the enabling technology that is driving the knowledge industry. As the development of software is mostly a design process, where new artefacts are conceived and built, the prediction of outcomes in the process is fraught with difficulties. Software project estimating is one of the essential Software Engineering techniques that will enable the rationalisation of decision-making regarding software development. Estimates that are more accurate will increase the probability of success and lower the risk. This thesis analyses the current software project estimating techniques available to practitioners and examines current practice in the estimating of software projects within the Western Australian industry.

The principal techniques examined are Function Point Analysis and COCOMO and these are shown to be flawed in their construction. The practices adopted by expert and experienced practitioners are analysed and it is shown that the formal algorithmic models are not widely used. It is also shown that estimates are required in a project's lifecycle before the full requirements are known. The Western Australian practices are also compared to similar analyses conducted in other countries.

Table of Contents

Table of Figures

Table of Tables

Chapter 1 Introduction

1.1 The Background to the Study

As society and the world economy moves into the second millennium, the service industries involving knowledge workers will continue to increase. As Quinn, Baruch & Zien (1997) p26-31 describe commodity prices have been falling steadily whereas the price of intellectually derived goods and services has been increasing. Software is the enabling technology that is driving the knowledge industry.

Computer systems are now ubiquitous. Computers impact on virtually all aspects of modern industrial society and are critical to the manner in which society and business operates. Computers are used to teach, educate, govern, manage, entertain and manufacture. Most electrical and mechanical equipment now includes computers, in part, to provide control and functionality. The enabling factor in computers is the software. Without software, computers are just a complex arrangement of minerals and metals. Therefore the effective functioning of modern society and business is becoming increasingly dependent on the production of cost effective software that is delivered in a timely manner.

As the development of software is mostly a design process where new artefacts are conceived and built the prediction of outcomes in the process is fraught with difficulties. Also the tools and methods used for this process are evolving rapidly which further compounds the difficulty in estimating as historical data may not be relevant to the existing environment. Software development projects tend to be at the top end of complexity in human endeavours. The development of software cannot be compared with most industries, where it is normal to produce the same type of products repetitively, because software development is a continuous design process. It is interesting to note that with most human activities that are new or novel in nature it is difficult to predict the outcomes.

As a consequence when undertaking a complex design process the risk is high and failures are frequent. Numerous authors within the software industry have referred to the failure to deliver the expected outcomes as the "software crisis". Parnas (1994) p286 advised he has been hearing the term for 25 years and says it is neither sudden nor short term and also uses the term "chronic illness". Pressman (1997) p17 prefers to call it a "chronic affliction" because the problems in the industry have been causing pain and distress for a long time and it appears they will continue indefinitely.

The construction of software systems is dynamic with a large number of variables affecting its outcome. Some of the variables are known and others are not when the most critical estimates are required to be made at project initiation. As a consequence software projects experience a high rate of failure because their success criteria is judged on highly suspect initial estimates. They constantly fail to meet their financial, schedule, effort, functional and quality targets. There is a school of thought, Thomsett (1991), that argues any reasonable sized development a project can only meet one or two of the above targets. Software engineering is a new field of human endeavour, its knowledge base is low on how to effectively measure the attributes and entities that contribute to the building of systems. The demands and the environment both in terms of the requirements expressed and the enabling technology are changing and evolving rapidly.

There is a need for methods to improve our ability to work in such an environment and increase the probability of being successful in the delivery of software systems. Estimating is one of the essential Software Engineering techniques that will enable the rationalisation of decision-making regarding software development. More accurate estimates will increase the probability of success and lower the risk. As Abdel-Hamid, (1993) p20 advises, processes and procedures are also required that provide a step-wise feedback mechanism to enhance the accuracy of estimates as the projects proceed.

This research provides an analysis of the current practice used in Western Australia to estimate software projects, and practitioners perceptions of the usefulness of existing techniques. A comprehensive survey has been conducted of expert and experienced practitioners from a wide range of organisations in order to obtain data and practitioners perceptions. The organisations cover a broad spectrum in both size and types of projects developed. An analysis of existing techniques has also been undertaken to determine their strengths and weaknesses.

1.2 The Significance of the Study

Software is a critical component of most technological products and organisations. Decisions relating to the development of new products or business services are dependent on the cost, duration and effort involved in the software component of the development.

Abdel-Hamid, (1993), Boehm (1981) and Pressman (1997) assert planning is one of the pivotal activities in the software development process and accurate estimates are a precursor to good planning.

Without adequate estimates management decisions in software projects cannot be made with any certainty and as Weinberg (1993) advises most of the crises in the industry can be attributed to an inability to manage. A key input into the management and planning process is an estimate of the cost, schedule and effort of the work to be performed.

1.3 The Purpose of the Study

This research aims to define current practice for estimating new and maintenance projects. This will enable future estimating models to be derived that take into account current practice and avoid weaknesses of existing models whilst capitalising on their strengths. The analysis also addresses issues relating to the initial project estimates. This area is seen as important as the initial estimate is the one where business decisions are made as to the project viability and provides the major input into project planning. It is also recognised that the estimates are only part of the input into the decision making process as other parameters need to be considered such as company image, acceptability of the project to the employees etc. Existing estimating techniques only appear useful when a comprehensive requirements or design has been defined. This is too late in the lifecycle for the initial decision making process.

This work will also test Hope's (1996) high level structure of an estimating model and the parameters that would cause the nominal estimate to be modified. Hope's model uses a work breakdown structure to define the activities of a software project.

1.4 Research Questions

The questions that this research tries to answer are: -

- Is the Western Australian software industry's estimating practices similar to counterparts in other parts of the world?
- What are the current estimating techniques employed by software professionals in formulating the initial estimates for both new development and maintenance projects in Western Australia?
- What elements need to be considered and estimated at project start-up?
- What is the degree of confidence or certainty of the various elements estimated?
- What parameters or cost drivers are used to modify nominal estimates and what degree of impact do they have on an estimate?
- What are practitioner's perceptions of the estimating techniques' strengths and weaknesses?
- When in a project's lifecycle are the first estimates required?
- Do organisations and practitioners follow generally accepted good practice and processes?
- Do the formal estimating techniques possess any flaws?

Chapter 2 Review of the Literature

2.1 Introduction

This chapter looks at the history and general classification of software estimating techniques and methods. A detailed examination of the more prevalent techniques is given. Other surveys of estimating practice are discussed in chapter 5.

It would appear that Boehm (1981) p329-341 was the first to categorise estimating techniques into algorithmic models, expert judgement, analogy, Parkinson, price to win, top-down and bottom-up. These techniques are described as follows.

- Algorithmic models are where a method uses one or a number of algorithms to produce various estimates of effort, duration and cost as a function of variables that are considered to affect the outcome. These variables are generally referred to as cost drivers. The normal form of an algorithmic model is:

  $$E = aS^b$$

  Where E is the estimate, S is the size, a is a productivity parameter and b is a scale parameter that accounts for economies of scale and complexity of the system under development. Boehm's Constructive Cost Model - COCOMO is one such example. The b parameter indicates that there are non-linear relationships in software development and this is evident in several models and supported by Banker & Kemerer (1989). However, Kitchenham (1992) provides arguments that for a single environment linear models are likely to be sufficient. This assertion is further supported by the work of Briand, El Amam, and Bomarius (1997) in the development of their hybrid method.

- Expert judgement uses the experience and value judgements of one or more experts. A formal technique in this area is Wide-Band Delphi.
- Analogy uses the actual costs of similar past projects and extrapolates them for new developments. This is a case based reasoning approach.
- Parkinson as described by Boehm (1981) is a design to cost method that limits the development to the available resources. This is from Parkinson's Law which states "Work expands to fill the available volume".
- Price to win is establishing estimates based on the price that is expected to be acceptable for gaining a contract.
- Top-down is where the overall cost is derived for the project and then divided into the different phases.
- Bottom-up or Work Breakdown (WBS) estimating decomposes a project into its individual activities and tasks which are estimated separately and them summed to produce the overall estimate.

More recently Humphrey (1995) has extended this list to include his own proxy-based technique (PROBE) and Putnam's Fuzzy Logic. Putnam & Myers (1992) do not elaborate the Fuzzy Logic technique, however they do provide some useful information that can be incorporated into an estimating database.

From the literature surveyed, for example see Rutherford (1995) p66-75, the most widely reported and used estimating techniques are Boehm's Constructive Cost Model - COCOMO and Function Point Analysis - FPA. These techniques are considered to be formal techniques because they have a well-documented model with repeatable processes and methods by which estimates are calculated. These techniques are discussed in more detail below. The other techniques such as estimating by analogy are not well described, however one such description by Shepperd & Schofield (1997) is also given below. This method, analogy, purports to be in wide use in the software industry however as it is not formally described its application would vary widely from practitioner to practitioner.

The evidence from Hihn and Habib-agahi (1991) and Wydenbach and Paynter's (1995) surveys regarding technique usage show that if an algorithmic technique is used then either FPA or COCOMO (or a derivative) has a very high probability of being the technique used. Matson, Barrett and Mellichamp (1994) p275 also state that "the most frequently cited measures are lines of code and function point analysis". Hence it is essential that these particular algorithmic techniques are understood and discussed in detail.

2.2 Function Point Analysis - Albrecht

Function Point Analysis (FPA) is a software estimating technique designed to measure the size of a software system in terms of the functions that are delivered to a user. Function Points were devised by Albrecht (Albrecht, 1979). Jones (1991) reports the goals set for this measure were that it: -

- dealt with the external features of the software that were important to the user,
- could be applied early in a product's lifecycle,
- could be linked to productivity and
- be independent of the coding language.

Various modifications have been made to Function Points including Symonds Mark II Function Point metric and Jones' Feature Points. Both of these techniques are discussed below. These modifications came about because of perceived weaknesses such as not accounting for the inherent complexity of the application eg algorithmic complexity. Dreger (1989) was instrumental in making this estimating measure available to the general public with his publication, which was essentially a function point tutorial. Garmus & Herron (1996) is probably the most recent publication that provides function point counting guidance which includes examples for the counting of Graphical User Interface applications.

Function Points measure software by quantifying the functionality provided to the user based primarily on logical design. The objectives of function point analysis are to: -

- Measure functionality that the user requests and receives
- Measure software development and maintenance independently of the technology used for implementation.

The function point count reflects the specific countable functionality provided to the user by the project or application. The application's specific user functionality is evaluated in terms of what is delivered by the application, not how it is delivered. Only user-requested and defined components are counted. The counts are conducted in two steps. Firstly an unadjusted or raw count is performed and then modified by applying factors that are specific to the project. These factors are termed Value Adjustment Factors VAFs. The unadjusted function point count has two function types - data and transactional. The composition of these function types are shown in Figure 2-1.

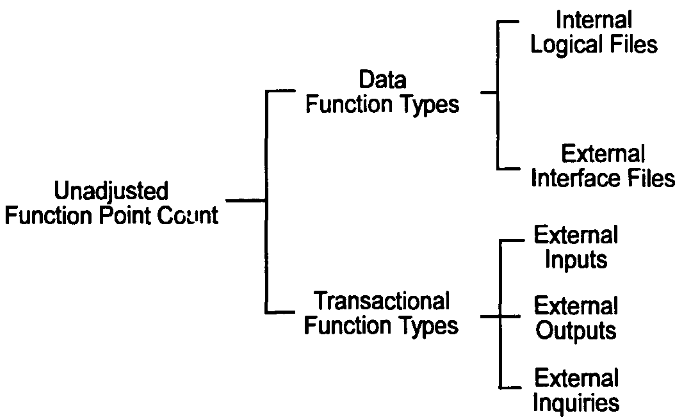

Figure 2-1 Composition of Function Types

Data function types represent the functionality provided to the user to meet internal and external data requirements. Data function types are either internal logical files or external interface files.

Transactional function types represent the functionality provided to the user to process data by an application in terms of inputs, outputs and inquiries.

The steps undertaken to count Function Points are as follows:-

- Determine the count type
- Determine the boundary of the application
- Identify all data and transaction functions
- Determine the unadjusted or raw function point count
- Determine the value adjustment factor
- Calculate the final adjusted function point count

These steps are covered in more detail below. A worked example is also given below.

1. **Determine the count type.** There are three types of function point counts which are:-

    - Development project function point count, which measures the functionality of a new application being developed and includes data conversion and capture components of the system.
    - Enhancement project function point count, which measures modifications by adding, deleting and changing an existing application.
    - Application function point count, which measures a system in production.

Formulae are given for each type in section 6 below.

2. **Determine the boundary of the application.** This is based on the user's definition of the scope of the system. Context diagrams and the appropriate Information Engineering techniques can be used here.
3. **Identify all data and transactional functions.** Data function types represent the functionality provided to the user to meet internal and external data requirements. Data function types are either internal logical files or external interface files.
   - An Internal Logical File (ILF) is a user identifiable group of logically related data or control information maintained by an elementary process within the boundary of the application being counted. The ILFs of an application are best represented by an Entity - Relationship (ER) model in third normal form. This, together with the attributes represents the logical model. The International Function Point User Group (IFPUG) definition of an entity is a Record Element Type (RET) and an attribute as a Data Element Type (DET). Any aspects relating to the physical implementation of the system are **NOT** taken into account. Care must be taken to ensure that only the entities and attributes that are user identifiable are counted. For instance, when resolving a many to many relationship the intersection entity could or could not be included in the count and is dependant on whether or not it is user identifiable. Foreign keys are counted as DETs if the user requires the relationship between the entities to be maintained. Multiple RETs would be represented as sub-types in the ER model.

- An External Interface File (EIF) is a user identifiable group of logically related data or control information used by the application but maintained **outside** the boundary of the application being counted. An EIF in one application must be an ILF in another. The identification and counting practices for an EIF are similar to the ILF above.

Transactional function types represent the functionality provided to the user to process data by an application. Transactional function types are defined as external inputs, external outputs and external inquiries.

- An External Input (EI) is the smallest meaningful activity that processes data or control information that comes from outside the boundary of the application being counted. The process must be self contained and unique. This includes all creating, modifying and deleting activities.
- An External Output (EO) is the smallest meaningful activity that generates data or control information sent outside the boundary of the application being counted. The process must be self contained and unique. This is typically in the form of reports, graphical displays, data transfer files and notifications.
- An External inQuiry (EQ) represents a combination of input (request) and output (retrieval) that results in retrieval of data from the application being counted. This typically involves a query on the application and can be in the form of look-ups, display and list boxes.

4. **Determine the unadjusted or raw function point count.** The raw function point count is calculated by determining the complexity of the data and transaction function types in accordance with the number of entities (RETs) and attributes (DETs) affected. The complexity is categorised as low, average or high. Contained in Table 2-1 is a summary of how the function point complexity ratings are ascertained.

The process is:-

- Each elementary component is counted and assigned a complexity value.
- The sum of each category is then multiplied by the complexity weighting.
- The categories are then totalled together to provide a raw or unadjusted function point count. This unadjusted function point count reflects the specific countable functionality provided to the user by the project or application. The application's specific user functionality is evaluated in terms of what is delivered by the application, not how it is delivered. Only user-requested and defined components are counted.

Note The complexity of an EQ has two components, input and output. Both must be determined and the higher result of the two used in the unadjusted function point count. The weightings are the same as those of EI and EO.

Input Complexity - EI	1-4 attributes	5-15 attributes	16+ attributes
0 or 1 files accessed	Low	Low	Average
2 files accessed	Low	Average	High
3 + files accessed	Average	High	High
Complexity weight : Low = 3, Average = 4, High = 6			

Output Complexity - EO	1-5 attributes	6-19 attributes	20+ attributes
0 or 1 files accessed	Low	Low	Average
2 or 3 files accessed	Low	Average	High
4 + files accessed	Average	High	High
Complexity weight . Low = 4, Average = 5, High = 7			

File Complexity - ILF	1-19 attributes	20-50 attributes	51+ attributes
1 logical record/entity	Low	Low	Average
2-5 logical records/entities	Low	Average	High
6+ logical records/entities	Average	High	High
Complexity weight : Low = 7, Average = 10, High = 15			

Interface File Complexity - EIF	1-19 attributes	20-50 attributes	51+ attributes
1 logical record/entity	Low	Low	Average
2-5 logical records/entities	Low	Average	High
6+ logical records/entities	Average	High	High
Complexity weight : Low = 5, Average = 7, High = 10			

Enquiry Input Complexity - EQ	1-4 attributes	5-15 attributes	16+ attributes
0 or 1 files accessed	Low	Low	Average
2 files accessed	Low	Average	High
3 + files accessed	Average	High	High
Complexity weight : Low =3, Average = 4, High = 6			

Enquiry Output Complexity - EQ	1-5 attributes	6-19 attributes	20+ attributes
0 or 1 files accessed	Low	Low	Average
2 or 3 files accessed	Low	Average	High
4 + files accessed	Average	High	High
Complexity weight : Low = 4, Average = 5, High = 7			

Table 2-1 Function Point Complexity Ratings

5. **Determine the Value Adjustment Factor**. In order to determine a final count for the system the raw count is modified by quantifying the key characteristics of the project and its environment and applying the resultant number to the raw count. These modifying characteristics are called the value adjustment factor

(VAF) which indicates the general functionality provided to the user of the application. The VAF is comprised of 14 general system characteristics (GSCs) that assess the general functionality of the application. See Table 2-2 for a listing of these 14 GSCs.

1.	Data communications	8.	Online update
2.	Distributed data processing	9.	Complex processing
3.	Performance	10.	Reusability
4.	Heavily used configuration	11.	Installation ease
5.	Transaction rate	12.	Operational ease
6.	Online data entry	13.	Multiple sites
7.	End-user efficiency	14.	Facilitate change

Table 2-2 FPA General System Characteristics

Each characteristic has six degrees of influence with associated descriptions that help determine the degree of influence of the characteristic. The degrees of influence range on a scale of zero to five as follows:

0 = not present or no influence;

1 = minor or incidental influence;

2 = moderate influence;

3 = average influence;

4 = significant influence;

5 = strong influence throughout.

The total VAF is determined by evaluating all fourteen general system characteristics and summing them to produce the Total Degree of Influence (TDI). The TDI is inserted into the following equation to produce the value adjustment factor.

$$VAF = (TDI * 0.01) + 0.65.$$

When applied, the value adjustment factor adjusts the raw function point count by plus or minus thirty five percent (± 35%) to produce a function point count.

6. **Calculate the final adjusted Function Point Count.** The final adjusted function point count is calculated using specific formulae for development projects, enhancement projects, or existing applications. These formulae are given below using the following abbreviations:-

DFP - Development project function point count.

EFP - Enhancement project function point count.

AFP - Application project function point count.

UFP - Unadjusted function point count.

CFP - Conversion unadjusted function point count.

ADD - Unadjusted function point count **added** by enhancements. Also used for an existing application.

CHGA - Unadjusted function point count of modified functions by enhancements **after** modification.

CHGB - Unadjusted function point count of modified functions by enhancements **before** modification.

DEL - Unadjusted function point count **before** the enhancements.

UFPB - Application unadjusted function point count **added** by enhancements.

VAF - Value adjustment factor.

VAFB - Value adjustment factor **before** enhancements.

VAFA - Value adjustment factor **after** enhancements.

DFP = (UFP + CFP) * VAF

EFP = [(ADD + CHGA + CFP) * VAFA] + (DEL * VAB)

AFP = ADD * VAF

Note that the application function point count after enhancement is calculated by:

AFP = [UFPB + ADD + CHGA) - (CHGB + DEL)] * VAFA

(International Function Point Users Group, 1994)

2.2.1 Function Point Analysis Example.

The following is a simple example of a function point count for a development project based on a very simplified project management system.

1. **Determine the count type**

   For this example, it is assumed that the project is a new development and hence the type is "Development project".

2. Determine the boundary of the application

Creating and analysing a context diagram such as the example in Figure 2-2 usually determines the boundary of the application of interest.

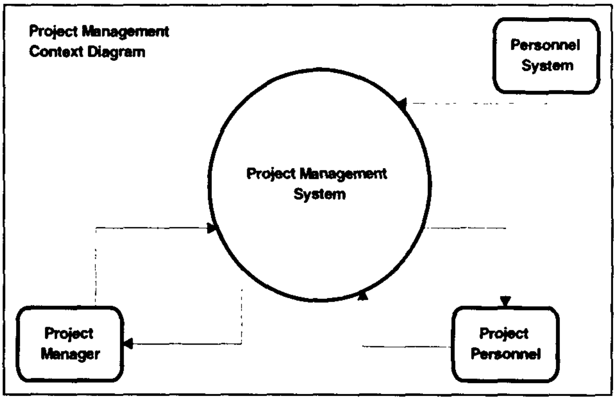

Figure 2-2 FPA Example - Context Diagram

3. Identify all data and transaction functions

An entity - relationship diagram is used to determine the logical files in the system and the attributes or data stored by the system also needs to be documented. Figure 2-3 provides for the model for our example. Note the personnel entity is maintained by the personnel system and thus is classified as an interface file.

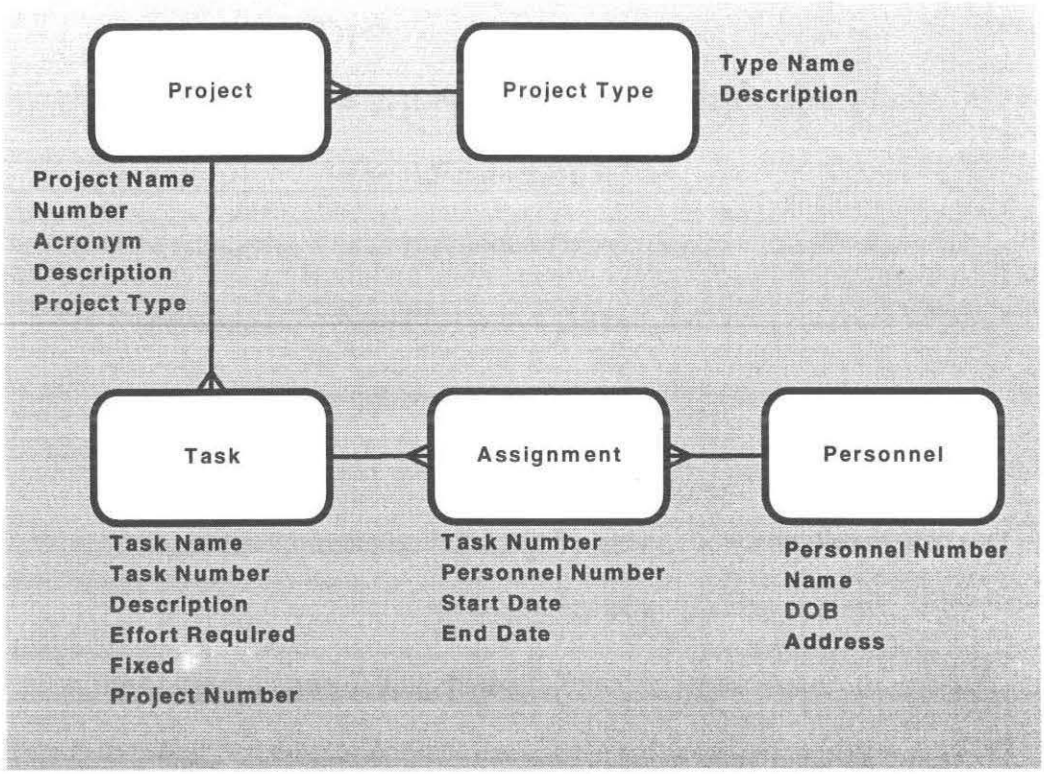

Figure 2-3 FPA Example E-R Diagram

Some functions required by the system could be:-

- Create Project
- Create tasks for a project
- Assign personnel to tasks
- Report all tasks on a project
- Personnel Inquiry

4. Determine the unadjusted or raw function point count

The unadjusted or raw function point count is achieved by considering the data function types and transactional functions in the system. The following are the calculations for the example:

Data Function Types

Project = an ILF with 5 DETs

Task = an ILF with 6 DETs

Assignment = an ILF with 4 DETs

Personnel = an ELF with 4 DETs

Project Type is not counted as this entity was created through normalisation and was **not** requested by the users. Therefore, we have three low complexity ILFs (3 * 7) and one low complexity EIF (1 * 5). These contribute **26** unadjusted Function Points to the count.

Transactional Function Types

There isn't sufficient information to count these function types accurately, therefore assumptions will have to be made. The results, for our example are detailed in Table 2-3.

Create Project	An EI with 5 DETs and one file referenced. Simple = 3
Create tasks for a project	An EI with 6 DETs and two files referenced (Project & Task). Average = 4
Assign personnel to tasks	An EI with 4 DETs and three files referenced. Average = 4.
Report all tasks on a project	An EO with 10 DETs and two files referenced. Average = 5
Personnel Inquiry	An EQ with 4DETs and one file referenced on the output side. Simple = 4
Total =	**20**

Table 2-3 FPA Example - Transaction Count

Therefore, the total unadjusted function point count for our example is achieved by summing the data function types and transactional functions. The result of this sum in our example is **46.**

5. Determine the value adjustment factor

The GSC ratings for the example are contained in Table 2-4.

1. Data communications	4	8. Online update	5
2. Distributed data processing	0	9. Complex processing	1
3. Performance	2	10. Reusability	0
4. Heavily used configuration	2	11. Installation ease	0
5. Transaction rate	2	12. Operational ease	3
6. Online data entry	5	13. Multiple sites	0
7. End-user efficiency	4	14. Facilitate change	4

Total Degree of Influence = 32

Table 2-4 FPA Example - VAF

Using the formula VAF = (TDI * 0.01) + 0.65

The Value Adjustment factor for this project = (32 * 0.01) + 0.65 or 0.97.

6. Calculate the final adjusted function point count.

Using the formula DFP = (UFP + CFP) * VAF.

The Development Function Point count = (46 + 0) * 0.97 or 45 Function Points (rounded).

2.3 Function Point Analysis Mark II

Symons (1988) who proposed an alternative estimating method, Function Point Analysis Mark II (FPA MK II), has critically examined Albrecht's FPA method.

Symons considers the FPA Mk II method to be an evolution of Albrecht's FPA and overcomes the following weaknesses:

- the classification of system component types,
- the lack of objectivity in choosing the complexity weights,
- the determination of internal processing complexity,
- the manner in which interfaces are counted and
- the restrictive nature and "degree' of influence" of the VAFs.

These weaknesses are discussed in greater detail in the chapter 6 - Analysis of Existing Estimating Models.

FPA Mk II is based on the premise that a system consists of logical transaction types with each transaction type being a logical input/process/output combination. Therefore the inputs, outputs, inquiries and even interfaces of Albrecht's FPA are all treated as a combination of input/process/output. Symons (1988) asserts that the concept of a logical file is "almost impossible to define unambiguously" and the correct concept to use is that of a logical entity. It should be noted that Symons (1988) refers to entities as "anything (object, real or abstract) in the real world about which the system provides information". Symons (1988) then discusses the Mark II model in the context of using an entity relationship data model. No stipulation as to the level of normalisation, of the data model, is given.

Another difference is that with Albrechts's FPA the "size" is measured as the *value* of the functions delivered to a user whereas Symons system size relates to the *effort* to develop the functions of the system.

In order to provide a process size measure of each transaction Symons (1988) considered the work of McCabe (1976) and Jackson (1975) to arrive at the hypothesis that a measure of processing complexity is to count the number of entities referenced by a transaction type. Referenced means any access to the entity - create, read, update or delete. The reasoning is that the access path through an entity model involves a selection or branch or loop. Therefore, the number of entities referenced by a transaction type is the measure of processing complexity. For other components of a logical transaction, input and output, the number of data element types are the measure of the size of the component. The formula for calculating Mark II Unadjusted Function Points (UFP) is:

$$UFP = N_I W_I + N_E W_E + N_O W_O$$

where

N_I = number of input data element types,

W_I = weight of an input data element type,

N_E = number of entity type references,

W_E = weight of an entity type reference,

N_O = number of output data element types,

W_O = weight of an output data element type.

It should be noted that N_I, N_E, N_O are each summed over all transactions.

The weights were determined by calibration using data taken from twelve existing projects to arrive at the average man-hours per component. These results were then scaled to make the Mark II technique compatible with Albrecht's. This compatibility ensured all eight systems, in the calibration data set, under 500 UFP's came out to be identical on both scales. These weights were:

$W_I = 0.44,$

$W_E = 1.67,$

$W_O = 0.38.$

The Mark II's Value Adjustment Factor (then known as the Technical Complexity Factor) utilises the fourteen factors proposed by Albrecht (see figure 3) with the addition five new ones. These new factors are for:

1. interfacing to other applications,
2. security features,
3. direct use by third parties,
4. special user training needs,
5. documentation requirements.

The technique also allows additional factors to be used by an organisation on the provision that the factors are only those that can be derived from user requirements.

2.4 Feature Point Analysis

Jones (1991) developed this technique in order to "give the benefits of the function point method to real-time software, embedded software, systems software and telecommunications software". This technique was designed to overcome the perceived weaknesses of the function point technique with algorithmically complex systems. The technique uses the average complexity weighting of Albrecht's technique and adds a new parameter - algorithms with weighting of three. In addition it reduced the weighting of the files parameter from ten to seven. The parameters and their weighting's are summarised in Table 2-5.

This technique is not a simple extension to include the algorithm parameter, as alluded to by Pressman (1997), but uses a totally different method to calculate complexity.

Parameter	Complexity Weight
Algorithms	3
Inputs	4
Outputs	5
Inquiries	4
Files	7
Interface Files	7

Table 2-5 Feature Point Analysis Parameters

Complexity is not adjusted by using the fourteen value adjustment factors but by answering two questions that Jones (1991) claims summarises their intent. These questions relate to the problem complexity and data complexity as follows:

Problem Complexity.

1. Simple algorithms and simple calculations?
2. Majority of Simple algorithms and simple calculations?
3. Algorithms and calculations of average complexity?
4. Some difficult algorithms and calculations?
5. Many difficult algorithms and calculations?

Data Complexity.

1. Simple data with few variables and low complexity?
2. Numerous variables but simple data relationships?
3. Multiple files, fields and data interactions?

4. Complex file structures and data interactions

5. Very complex file structures and data interactions?

Both questions are answered and the resultant number summed together. Then a complexity multiplier as detailed in Table 2-6 is applied to the unadjusted function point count to obtain the final count.

Sum of Problem & Data Complexity	Complexity Multiplier
2	0.6
3	0.7
4	0.8
5	0.9
6	1.0
7	1.1
8	1.2
9	1.3
10	1.4

Table 2-6 Feature Point Complexity Multipliers

Jones (1991) asserts that Feature Points returns the same adjusted function point count as does Albrecht's techniques and covers the same range but in a much simpler fashion. Note there are discrepancies in the way the Feature Point technique is described by Jones (1991). That is on p111 he discusses three questions that must be answered however on p112 and p318 only the two questions described above are used.

2.5 COCOMO

COCOMO stands for COnstructive COst MOdel. COCOMO was first described by Boehm (1981) and comprises three models that correspond to available information at different stages in the development process. Each of these models includes a number of algorithms relating product size in thousand lines of delivered source instructions (KDSI) to the development effort in months (MM_{nom}). COCOMO's three models are:

- basic COCOMO for initial estimates;
- intermediate COCOMO for when the major subsystems are determined and
- detailed COCOMO when individual modules within the subsystems have been identified.

The models' effort equations are of the form

$$MM_{nom} = a(KDSI)^{b}$$

where effort is measured in person months and size is measured in thousands of delivered source instructions (KDSI). The values of *a* and *b* depend on the model being used and the mode of development. See Table 2-7.

Mode	**Basic**		**Intermediate & Detailed**	
	a	b	a	b
Organic	2.4	1.05	3.2	1.05
Semi-detached	3.0	1.12	3.0	1.12
Embedded	3.6	1.20	2.8	1.20

Table 2-7 COCOMO Coefficients

These modes are Organic, Semi-detached and Embedded, which represent increasingly complex software development projects.

Organic is used to describe the situation of relatively small teams developing software in a highly familiar in-house environment. Most people connected with the project have extensive experience working with related systems and the requirements and schedule are not rigorously defined. The development environment is stable with little changes to existing operational hardware and procedures.

The Semi-detached mode is a mid-point between the extremes of Organic and Embedded. The team members have an intermediate level of experience with related systems and there is a mixture of skilled and unskilled people. The requirements and schedule are more rigorously defined than the Organic mode.

The Embedded mode is used for projects that need to operate with tight constraints. The resultant product must operate within a strongly coupled complex of hardware, software, regulations and operational procedures. An Embedded mode project tends to operate in new areas of application, hardware and development environments.

The coefficient values and the cost drivers described below were determined by expert opinion and a database of sixty-three projects was used to refine the values. Table 2-8 summarises these cost drivers. Note that all ratings categories are not applicable for each cost driver.

Fifteen cost drivers are used to modify the basic equation for intermediate and detailed COCOMO by means of multipliers. These cost drivers are categorised into product, process and resource attributes. The level of each cost driver must be assessed on a six point ordinal scale.

The basic effort estimate MM_{nom} is adjusted by the product of all the cost driver multipliers.

Cost Drivers	Description	Ratings					
		Very Low	Low	Nominal	High	Very High	Extra High
RELY	Required software reliability	0.75	0.88	1.00	1.15	1.40	
DATA	Data base size	-	0.94	1.00	1.08	1.16	-
CPLX	Product complexity	0.70	0.85	1.00	1.15	1.30	1.65
TIME	Execution time constraint	-	-	1.00	1.11	1.30	1.66
STOR	Main storage constraint	-	-	1.00	1.06	1.21	1.56
VIRT	Virtual machine volatility	-	0.87	1.00	1.15	1.30	-
TURN	Computer turnaround time	0.79	0.87	1.00	1.07	1.15	-
ACAP	Analyst capability	1.46	1.19	1.00	0.86	0.71	-
AEXP	Applications experience	1.29	1.13	1.00	0.91	0.82	-
PCAP	programming capability	1.42	1.17	1.00	0.86	0.71	-
VEXP	Virtual machine experience	1.21	1.10	1.00	0.90	-	-
LEXP	Programming language experience	1.14	1.07	1.00	0.95	-	-
MODP	Use of modern programming practices	1.24	1.10	1.00	0.91	0.82	-
TOOL	Use of software tools	1.24	1.10	1.00	0.91	0.83	0.77
SCED	Required development schedule	1.23	1.08	1.00	1.04	1.10	-

Table 2-8 COCOMO Cost Drivers

The important points about Intermediate and Detailed COCOMO are not just the introduction of the cost drivers. Intermediate COCOMO is intended to be used when the major components of the software product have been identified. This enables effort estimates to be made on a component basis using the size and cost driver ratings appropriate for each component. The adjusted component estimates are summed to attain the total estimate. Detailed COCOMO takes the estimation process further and uses cost driver multipliers that differ for each major development phase. COCOMO also has features for handling adapted code and assessing the maintenance effort. Code re-use effects are determined by calculating an equivalent number of delivered source instructions (EDSI), and using EDSI in place of DSI in the effort equations. Maintenance effort estimates are restricted to that which is expended on the following:

- redesign and development of small portions of a product;
- design and development of small interface packages that require some redesign of the product;
- modification of the software's code, documentation or database structure.

The Basic COCOMO estimate for annual software maintenance is calculated in terms of the annual change traffic (ACT) which is the fraction of the software product's source instructions that undergo change during a year. It is calculated using the following equation:

$$MM_{AM} = ACT * MM_{nom}$$

where

MM_{AM} is the estimated annual maintenance effort;

MM_{nom} is the estimated development effort.

Boehm (1981) suggests that the annual maintenance estimate can be refined by using the Intermediate COCOMO cost drivers with the following adaptations.

- SCED is not used.

- Personnel ratings and computer turnaround are related to the maintenance staff and computer.
- New cost driver multipliers are used for RELY and MODP.

COCOMO uses a relationship between the development time (schedule) and development effort using the following equation;

$$TDEV = a(MM)^b$$

where

TDEV is the development time in months;

MM is the estimated effort to produce the product in man-months;a and b are constants that depend on the mode of development as shown in Table 2-9. The same values are used for Basic, Intermediate and Detailed COCOMO.

Mode	a	b
Organic	2.5	0.38
Semi-detached	2.5	0.35
Embedded	2.5	0.32

Table 2-9 COCOMO Schedule Equations Coefficients

The COCOMO model also defines details such as a man month consists of 152 hours of working time and perhaps most importantly provides a phase and Work Breakdown Structure (WBS) for which the model applies. Boehm (1981) also details assumptions such as the project "enjoys good management" and "the requirements specification is not substantially changed after the requirements phase". Boehm's work is thorough and demonstrates an excellent understanding of the realities of software development.

Boehm (1987) also developed an improved version of COCOMO which is based on a more modern process model which includes risk management and can be used to predict the costs of Ada projects.

2.6 COCOMO 2.0

COCOMO 2.0 is currently under development and as yet there are only unpublished preliminary manuals available. This work will be very important and impact on all future software estimating models. It was recognised by Boehm, Clark, Horowitz and Westland (1996) that COCOMO had increasing difficulty in estimating the costs and schedules of business software, object oriented software, software developed using an evolutionary approach and software that is a composite of commercial packages.

An anticipated model of future software development practices has guided COCOMO 2.0's construction. This model's components are outlined below.

- **End-user programming** - where applications will be developed using application generator tools such as spreadsheets, query systems and parameter driven specialised systems.
- **Infrastructure** - where applications will be in the areas of operating systems, data-base management systems and networks operating systems together with the user interface tools.
- **Application Generators** - where the bulk of the tools used by the end users will be developed such as financial analysis tools, project management tools, etc.

- **Application Composition** - where applications too complex for a single tool will be created from several inter-operable components.
- **Systems Integration** - where large scale, embedded or unusual systems will be developed that require a significant amount of customised software development.

COCOMO 2.0 provides a suite of increasingly detailed estimation models in order to satisfy the different practices. End user practice is not seen by Boehm et al (1996) to need a COCOMO 2.0 model as the applications are simple and will be developed in a small number of days. The first model addresses the Application Composition practice that comprises applications that cannot be built using a specific tool such as a spreadsheet. However, the application can be created using a number of diverse packages. The approach used is called Object Point estimation. This technique is similar to Function Point analysis in that it uses a like process that is outlined below.

1. Assess object counts: estimate the number of screens, reports and 3GL components that comprise the application.
2. Classify each object instance into simple, medium and difficult complexity levels using supplied tables.
3. Assign a weight to each instance using a supplied table.
4. Add all the object instances to obtain an Object Point count.
5. Estimate the percentage of re-use expected to be achieved in the project using the following formula:

$$\text{New Object Points} = \frac{(\text{Object Points}) * (100 - \%\ \text{Re-use})}{100}$$

6. Determine a productivity rate (productivity being measured in terms of the New Object Points per person month) from the supplied table.
7. Compute the estimated person months.

The second and more detailed model, Early Design, uses unadjusted Function Points as a sizing metric. The VAFs are not used as COCOMO (1995) advises that the characteristics and relative weighting are inconsistent with their experience. The unadjusted Function Points are translated into source lines of code (SLOC) and then KSLOC by using tables such as those provided by Jones (1991). A set of cost drivers is then applied.

The third model, Post Architecture uses KSLOC as per the Early Design model but uses a more comprehensive suite of cost drivers. This model is only for use after the project's architecture has been defined.

The importance of standard sizing measures has been recognised by the authors of COCOMO 2.0. The counting rules for the source lines of code are based on the Software Engineering Institute's source statement definition checklist. The function point counts are based on the IPUG's guidelines.

2.7 Expert Judgement

The techniques in this area involve consulting with experts to obtain their opinion and consequent estimate as to the effort, cost and schedule factors for a particular project. An expert can factor in elements of a project such as the skill of the people involved, the similarity with past projects and political aspects of the development.

If a single expert's opinion is obtained then the result can be subject to bias and unfamiliarity with major aspects of the system.

To overcome the difficulties associated with a single expert an number of group consensus techniques have evolved such as the Delphi technique. This technique originated at the RAND Corporation and Boehm's Wideband Delphi version is described by Boehm (1981).

The use of the Wideband Delphi technique proceeds as follows.

1. A coordinator provides each expert with a specification of the system and an estimation form.
2. A group meeting is held in which the project and estimation issues are discussed.
3. The experts form an estimate individually and anonymously including rationale they feel may be required.
4. The coordinator summarises all the estimates and distributes to all the experts without the rationale.
5. Another group meeting is held which focuses on the areas where there is a wide divergence of opinion. These areas are discussed in depth to ensure all experts have an understanding of the issues involved.
6. The experts make another estimate individually and anonymously and steps 4 to 6 are iterated to obtain convergence.

This method ensures that there is good understanding of all the issues involved through communicating at the meetings whilst also minimising the impact of any dominant individual.

This technique has been extended by Hope (1997) whereby detailed estimating forms are provided to the experts that require them to make optimistic, probable and pessimistic estimates of both cost and effort. The elements of the forms were derived from analysing five large projects implemented on a national basis within Telecom Australia. The method has not been validated however it has proved useful to identify cost and effort factors not considered by other known techniques. For instance in one project (known to the author when a IT Manager in a national corporation) with a total cost of $4.8m, $1.3m was identified to environmental costs. A formula

$$Estimate = \frac{Optimistic_{Tot} + (4 * Probable)_{Tot} + Pessimistic_{Tot}}{6 * E_{Tot}}$$

is used to give a weighting to the sum of the estimates. E_{Tot} is the number of experts providing estimates. The rationale behind the equation is the standard deviation of a beta distribution.

2.8 Analogy

Shepperd & Schofield (1997) describe the only analogy based estimating model found in the literature. The basis of their technique is to describe by way of a number of variables the project to be estimated and then use this description to find similar completed projects. The known values of the completed projects can then be applied to form an estimate for the new project. The similarity of projects is defined in terms of project features such as number of interfaces, development method etc and can vary in number. Similarity is defined as proximity in n-dimensional space and the authors use an unweighted Euclidean distance. Unfortunately, the authors do not provide a full description of their technique.

2.9 Other Techniques

There are numerous other estimating models available and work is being conducted to extend and enhance existing models. The one of most significance is Full Function Points that is described below.

- Full Function Points - This an extension of the IFPUG Function Point standard to cater for the criticisms levelled relating to its inability to cope with real time software. Full function points has been designed for both management information systems and real time software. The IFPUG counting rules dealing with control aspects have been expanded and new function types added to address the control aspects of real time software. These consist of two control data function types - Updated Control Group - UCG and Read-only Control Group - RCG. There are also four new control transactional function types - External Control Entry - ECE, External Control Exit - ECX, Internal Control Read - ICR and Internal Control Write - ICW.

  (St-Pierre, Maya, Abran, Desharnais, and Bourque 1997)

There were no models found in the literature search that used a detailed workbook or work breakdown approach except that proposed in Hope (1996).

Chapter 3 Theoretical Framework

There are several areas of theory that relate to this work. These are:

- measurement theory that relates to both analysing the validity of existing estimating techniques and the construction and evaluation of the survey instrument;
- theory supporting surveying and sampling is important in that sample size, bias and an understanding of the respondent profile affects the conduct of the survey and the response analysis. This also provides a foundation on which to critically analyse other survey efforts within the domain of this work.
- theory relating to the statistical treatment of the survey data.

3.1 Measurement Theory

3.1.1 Measures

As Lederer and Prasad (1998) assert the estimating of software projects have important ramifications for organisations that are making decisions based on the estimates and also on the teams and personnel who undertake the projects. Therefore it is important that any measures derived for estimating purposes must be based in measurement theory if they are to have any mathematical validity and hence meaningfulness. It is apparent that a number of "metrics" in the Software Engineering paradigm fail to take heed of the available theory and hence the metrics espoused are flawed (Fenton 1994). More so the conclusions drawn from metrics must be treated with suspicion.

Fenton (1994) defines measurement as "the process by which numbers or symbols are assigned to attributes of entities in the real world in such a way as to describe them according to clearly defined rules". An entity can be either an object, such as a requirements specification, or process of interest, such as the requirements phase of a project. An attribute is a property of an entity such as the number or words in the requirements document. There are two types of measurement, direct and indirect. Direct measurement is where the measurement of an attribute does not depend on the measurement of any other attribute. Indirect measurement is an attribute that comprises the measurement of one or more other attributes.

Hence it is important to note that measurement is a *defined mapping* of numbers or symbols to an attribute which must preserve any intuitive or empirical observations about the attribute.

For instance, we could measure the length of a requirements document by mapping to the attribute *length* the number of pages or the number of words comprising the document. To be clear about the attribute we would have to have a formal definition or model for the requirements document that defined the rules under which the measurement took place in order that the length could be stated unambiguously and in a repeatable fashion. For instance a requirements document model would have to cater for various aspects that could impact on the attribute such as page size, font size, line spacing, standard contents, etc. It is interesting to note where common measurements are taken this definition applies. For instance, in the measurement of the height of a person rules apply as to the person's attitude, ie standing with feet on the ground and the disposition of the footwear before mapping the person's length to a number system.

Fenton (1991) is of the opinion that where no previous measurement has been performed or the attributes are not well understood one should attempt to obtain direct measures in order to gain an understanding of the entity and attributes in question.

For measures to be valid it is generally considered that they should obey the representation condition of measurement theory (Fenton, 1991). The representation theory of measurement has a mathematical framework based on sets, relations, axioms and functions. The components are:

- Empirical relation systems that determine the axioms that characterise any empirical observations or relations between the entities. The set of entities E, together with the set of empirical relations R, is called an Empirical Relation system (E,R) for the attribute under observation. For example, the attribute length of a document leads a binary relation "is longer than" and this satisfies the axiom of transitivity. That is, if document A is longer than document B which in turn is longer than document C, then we may infer that A is longer than C. Relations do not have to always be binary, for instance, "is long" would only apply to an single instance of a document.
- The representation condition is required for measurement in order that the attribute defined in the E,R system can have a mapping M into a numerical

relation system (N, P) in such a way that all empirical relations are preserved. That is M maps attributes in E to numbers in N and empirical relations in R are mapped to numerical relations in P. Note the representation condition asserts that the correspondence between empirical and numerical relations is two way. For instance with the document example above if we considered E as the set of all documents and R contains the relation "longer than". Then a measure M of length would map E to the set of positive integers and "longer than" to the relation "> ". The representation condition asserts that document A is longer that document B, if and only if M(A) > M(B).

- The scale types that can be meaningfully applied to the measurement of an attribute are dependant on the representation mapping ***M*** from an empirical relation system ***E*** to some numerical relation system ***N***. If such a representation exists then the triple (***E, N, M***) is called the scale.

It should be noted that empirical relations are normally established by subjective means as a precursor to more objective forms. That is firstly observations are made of objects of interest in the real world in a subjective manner as a precursor to finding more objective measures. An example would be the measurement of temperature. It was no doubt observed in the past, before thermometers were invented, that the air temperature varied over time. The degree of hotness or coldness is subjective based on an individual's physiology and perceptions. At some stage the length of a column of mercury was mapped to a number system thus providing an objective measure not dependent on any individual' physiology or perception.

A framework for the validation of soft· are measurement has been proposed by Kitchenham, Pfleeger and Fenton(1995) which is useful in this work. The framework is based on Fenton's work and has the goals of helping both the areas of research and practice by facilitating the understanding of:

- measure validation
- validation work assessment
- appropriateness of measures in a given situation.

A structural model of measurement was provided and this can be seen in Figure 3-1.

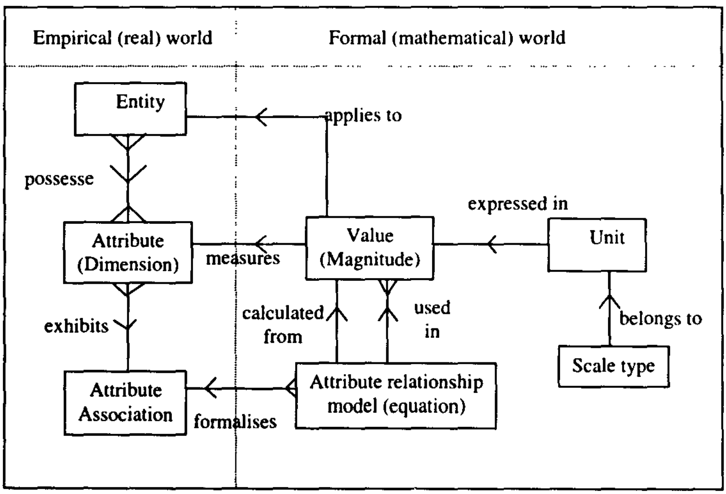

Figure 3-1 A structural model for measurement.

3.1.2 Scales

There are several distinct levels of measurement or scale types that variables can be mapped to. The allowable operations and hence statistics on the variables are dependent on the level of measurement achieved. These scale types are nominal, ordinal, interval and ratio and are described below.

Nominal

This is the weakest level of measurement that uses names, symbols or numbers to classify the characteristics in question. That is a nominal scale is used to label or categorise entities of interest. It should be noted the categories are qualitatively different not quantitatively. An example of a nominal scale is the categorisation of project activities into:

1 = management;

2 = analysis;

3 = design;

4 = coding;

5 = testing.

Hence the scaling operation is the partitioning of the set of entities of interest into a set of mutually exclusive subsets. The only relation is that of equivalence. In the example above the assignment of numbers is purely arbitrary and the classification scheme could have used a different set of numbers or letters such as:

ma = management;

an = analysis;

de = design;

co = coding;

te = testing.

See Table 3-1 for the summary of permissible operations and appropriate statistical tests.

<table>
<tr><th>Scale</th><th>Defining Relations</th><th>Examples of Appropriate Statistics</th><th>Appropriate Statistical Tests</th></tr>
<tr><td>Nominal</td><td>1. Equivalence</td><td>Mode
Frequency
Contingency coefficient</td><td rowspan="2">Nonparametric statistical tests</td></tr>
<tr><td>Ordinal</td><td>1. Equivalence
2. Greater than</td><td>Median
Percentile
Kendall r
Spearman r_s
Kendall W</td></tr>
<tr><td>Interval</td><td>1. Equivalence
2. Greater than
3. Known ratio of any interval</td><td>Mean
Standard Deviation
Pearson product- moment correlation
Multiple product-moment correlation</td><td rowspan="2">Nonparametric and parametric statistical tests</td></tr>
<tr><td>Ratio</td><td>1. Equivalence
2. Greater than
3. Known ratio of any two intervals
4. Known ratio of any two scale values</td><td>Geometric mean
Coefficient of variation</td></tr>
</table>

Table 3-1 Summary of measurement scales and statistics (Derived from Fenton (1991) p 36.

Ordinal

This is the second level of measurement that uses names, symbols or numbers to classify the characteristics of the entities in question and to describe a relationship between them. Typical relationships are greater confidence, more difficult etc. There is a "greater than" relationship between the categories and this must hold true for all pairs of categories. That is an ordinal scale is used to label or categorise entities of interest and order them by rank on a continuum. An example of an ordinal scale would be the categorisation of project activities into:

1 = very easy;

2 = easy;

3 = difficult;

4 = very difficult.

This scale does not provide any information regarding the distances between values and hence the interval between any pair of the categories may be larger or smaller than the interval between any other pair of categories. The example above could have been designated:

1 = very easy;

20 = easy;

300 = difficult;

4000 = very difficult.

Hence, the scaling operation is the partitioning of the set of entities of interest into a set of mutually exclusive subsets together with a ranking. The relations are that of equivalence and greater than. See also Table 3-1 for the summary of permissible operations and appropriate statistical tests.

Interval

An interval scale is the third level of measurement and has all the characteristics of an ordinal scale (the entities in question are classified and ranked) and in addition the distances between the entities can be measured. That is the mapping of the classification of the entities is of such precision that intervals between all the classes on the scale are known and represent equal quantities. The interval scale has a constant unit of measurement that assigns a real number to all pairs of objects in the set. In an interval scale the zero point and the unit of measurement are arbitrary. An example of an interval scale is the measurement of temperature using the Centigrade and Fahrenheit scales. The unit of measurement and the zero point are different however both scales contain the same information. As the scales contain the same information a reading on one scale can be transformed to an equivalent reading on the other. For instance the table below shows such a transformation and that the zero points and intervals are arbitrary. The freezing of water takes place at 0 degrees on the Centigrade scale and at 32 degrees on the Fahrenheit one.

Centigrade	0	10	30	100
Fahrenheit	32	50	86	212

This scale type is the first quantitative scale and arithmetic operations can apply as the ratio of the differences between the intervals is equal. Hence the numbers assigned can be added or subtracted and multiplied by a constant. The numbers cannot be multiplied or divided, as this scale type does not possess a true zero point. Thus statements such 40° C is twice as hot as 20° C are not valid.

See also Table 3-1 for the summary of permissible operations and appropriate statistical tests.

Ratio

A ratio scale is the fourth level of measurement and has all the characteristics of an interval scale (the entities in question are classified, ranked and has known equal intervals) and in addition has a true zero point. The ratio of any two scale points is independent of the unit of measurement. The unit of measurement is arbitrary. Examples are the measurement of length, weight and absolute temperature. For instance the weight of two objects can measured using grams and ounces and it is found that the ratio of the gram weights is identical to the ratio of the ounce weights. For temperature measured using the Kelvin scale which has a true zero point it is valid to say 40° K is twice as hot as 20° K albeit very cold.

Arithmetic operations apply to the objects as well as the intervals between the objects providing no negative values are used to multiply measurements.

See also Table 3-1 for the summary of permissible operations and appropriate statistical tests.

(Siegel, 1956 p22-29; see also Judd, Smith and Kidder 1991 p61-64)

3.2 Surveying and Sampling

One method by which the objects and events of the area of interest can be observed and data gathered is by the descriptive or normative survey method. (Leedy 1993). This method attempts to:

- describe the events that are currently taking place through data collected;
- draw conclusions from the data;
- extrapolate the conclusions into the general population.

This method assumes that what is observed is normal and under the same conditions could be observed again. That is, the phenomenon being observed follows a pattern that is common throughout the area in question and is normal.

The characteristics of a descriptive survey are:

1. The situation being observed needs to use this method as the means of data collection.
2. The population is carefully chosen to ensure a representative and valid sample is chosen.
3. The influence of bias in the data collected is minimised, acknowledged and accounted for in the data analysis.
4. The data collected is systematically organised in such a manner that valid and accurate conclusions can be drawn.

3.2.1 Population Selection Methods

One of the most important areas in surveying is the method used to select a representative sample. The sample size selected should have a probability that is great enough so that the resultant sample can be considered representative of the population under study.

There are two basic methods for sampling and these are non-probability and probability sampling.

Non-probability Sampling

With non-probability sampling there are two basic categories:

- convenience or accidental sampling and
- quota sampling.

Convenience or accidental sampling makes no effort to represent the population being studied. It takes data from the nearest convenient source and there is no attempt to control bias. An example would be a university lecturer wanting to make some generalisation about students, studies the students in his or her classes.

Quota sampling is a variant of convenience sampling and is also sometimes misleadingly referred to a "representative" sampling. With this type of sampling the objective is to select a sample that is replica of the population that one wishes to generalise and hence is "representative". With the sample parameters defined the subjects in the population are selected at convenience. For instance if a university lecturer was studying a student population that had equal numbers of females and males the instructions may be to interview equal numbers entering a certain building on campus. Bias will enter by several means. The students entering say a Computer Science building may have different characteristics than those entering an Art's building. The interviewer may interview friends, avoid dangerous looking or unattractive people and hence bias the data collected.

With non-probability sampling there is no way to estimate the probability that each element has been included in the sample and no assurance that every element has an equal chance of being included.

(Leedy 1993 p197-200 and Judd et al 1991 p130-136).

Probability Sampling

With probability sampling the probability that each element will be included in the sample can be specified and sufficient cases are selected from the population to ensure the validity of the mathematical operations applied to the data collected. As

Judd et al (1991) p133 advise "Probability sampling is the only approach that makes possible representative sampling plans".

With probability sampling the subjects are chosen from the population by means of a randomization method and hence the sample is known as a random sample. There are different methods employed in selecting a random sample some using a lottery method or a table of random numbers. The overriding principle is to permit blind chance to determine the outcomes of the selection process. For example if a university lecturer was studying a student population that had equal numbers of females and males he or she may use a random number generator to select students based on their identification number. In this manner each student would have the same probability in being selected and hence any individual bias is reduced. There are refinements to simple random sampling such as stratified random sampling and proportional stratified sampling which are not relevant to this work.

Sample Size Selection

It should be noted that when you sample you are dealing with only partial information and hence a risk exists of being wrong when inferring something about the population on the basis of sample information. This amount of risk relates directly to the size of your sample. The risk is mitigated a the sample size increases. To avoid the risk the entire population would need to be studied.

In selecting the sample size a number of factors need to taken into account. These are:

- the method of sampling to be used;
- the confidence level

- the precision (or reliability) range.

To minimise risk a high confidence, eg 95 percent, that the true value sought (the actual value in the population) lies somewhere within a small interval (say + or - 5 percent) around the sample value (your precision). Ross (1996) provides a number of formulas for determining sample size. For this work the following formula is appropriate.

$$n = \frac{P((1-P)}{\frac{A^2}{Z^2} + \frac{P(1-P)}{N}}$$

Where **n** = sample size required

N = number of subjects in the population

P = estimated percentage of the population possessing the attribute of interest

A = Accuracy desired (expressed as a decimal)

Z = number of standard deviation units of the sampling distribution corresponding to the desired confidence level.

3.2.2 Bias

Bias is inherent in all research however as Leedy (1993) p213 says "data in descriptive survey research are particularly susceptible to distortion through the introduction of bias into the research design". Leedy (1993) goes on to define bias as any influence or set of conditions that distort the data from that which would have been obtained by pure chance. Therefore the objective is to reduce the sampling errors and acknowledge and make explicit any possible source of bias. Using a random sampling method can reduce sampling errors.

The use of volunteers in survey research represents a major source of bias. Volunteers, as a group, possess characteristics quite different from those who do not generally volunteer. These differences need to accounted for when choosing to use an exclusively volunteer sample as the bias introduced into the data may be so great that there is little confidence in extrapolating the survey's findings to the population in general.

Research findings exist which describe several unique characteristics of the volunteer subject. By using these characteristics suitably, inadvertent biases usually associated with using and interpreting results from volunteer samples may be avoided. The following list provides a number of conclusions about the unique characteristics of the volunteer. The categories representing the higher levels of confidence to be placed in the findings are listed. Within each category, the conclusions are listed in order starting with those having the strongest evidence supporting them. (Rosenthall and Rosnow, 1975; p195-196):

Conclusions Warranting Maximum Confidence

Rosenthall and Rosnow, (1975) also advise:

- Volunteers tend to be better educated than non-volunteers, especially when personal contact between investigator and respondent is not required.

- Volunteers tend to have higher social-class status than non-volunteers, especially when social class is defined by respondents' own status rather than by parental status.
- Volunteers tend to be more intelligent than non-volunteers when volunteering is for research in general, but not when volunteering is for somewhat less typical types of research such as hypnosis, sensory isolation, sex research, small-group and personality research.
- Volunteers tend to be higher in need for social approval than non-volunteers.
- Volunteers tend to be more sociable than non-volunteers.

Conclusions Warranting Considerable Confidence

- Volunteers tend to be more arousal-seeking than non-volunteers, especially when volunteering is for studies of stress, sensory isolation, and hypnosis.
- Volunteers tend to be more unconventional than non-volunteers, especially when volunteering is for studies of sex behaviour.
- Females are more likely than males to volunteer for research in general, more likely than males to volunteer for physically and emotionally stressful research (eg., electric shock, high temperature, sensory deprivation, and interviews about sex behaviour).
- Volunteers tend to be less authoritarian than non-volunteers are.

- Jews are more likely to volunteer than Protestants, and Protestants are more likely to volunteer than Roman Catholics are.
- Volunteers tend to be less conforming than non-volunteers when volunteering is for research in general, but not when subjects are female and the task is relatively "clinical" (eg. hypnosis, sleep, or counselling research).

As can be seen that not all these attributes are relevant to the population in this study however serve to emphasise the wide nature of the characteristics of volunteers and how this could easily bias the data collected.

Borg and Gall (1979) have suggested how to use the characteristics of the volunteers empirically found to combat the effects of bias in survey research. For example, they suggest that "The degree to which these characteristics of volunteer samples affect research results depends on the specific nature of the investigation". For example, a study of the level of intelligence of successful project managers in different organisations would probably yield spurious results if volunteer subjects were studied, since volunteers tend to be more intelligent than non-volunteers. On the other hand, in a study concerned with cooperative behaviour of people in a project team situation, the tendency for volunteers to be more intelligent may have no effect on the results, but the tendency for volunteers to be more sociable could have a significant effect.

It is evident that the use of volunteers in conducting descriptive survey research confounds the interpretation of research data and its extrapolation to the population in general which of course includes all individuals who did not volunteer.

Other areas of bias may enter the data in the generalisations made from the responses in that a sub-group of the population in question may have motives that lead them not to respond. Leedy (1993) p214 gives as an example where misleading generalisations could easily be drawn from people responding to a question regarding taxation. The non-respondents may not have wished to make known some level of indiscretion that would lead to the surveyor drawing conclusions about their level of integrity.

3.3 Rating Scales

There are several methods to extract data from a respondent. One of the more accepted ways is to use a question that has an associated graphic scale to measure the strength or intensity of a respondent's feeling or attitude to the question. These types of questions allow the respondent to have finer granularity in his or her answer and hence more quantitative information can be obtained. One of the major advantages of a graphic scale is its ease of use.

Judd et al (1991) p 153 report that "self-ratings have shown to be equal or superior to other types of assessments in predicting a wide range of criteria". However as the ratings given are subjective errors and bias can be present. Some examples are:

- respondents tend to avoid extreme positions,
- an overall positive or negative attitude can influence specific question ratings,
- respondents tend to generosity in which overestimates are made of desirable qualities.

These issues can be addressed by constructing the questions in such a manner as to avoid extreme positions and have questions that check and balance the respondents' attitude.

Judd et al (1993) p154-155 warn that it is a common fallacy to interpret a rating scale response literally. For example eighty percent of responses may be above the average category although this is impossible by definition. The rating information is implicit in the relationships to other measures.

Judd et al (1991) report that Masters (1974) advises the number of categories on the scale can influence reliability and also indicates "five to seven categories seems to limit reliability, although increasing the number of categories over this number helps little if at all".

The most common and easily used graphic intensity (or scaled) question involves the use of the Likert-type answer scale. It allows the respondent to choose one of several (usually five to seven) degrees of attitude about a statement. Summing the items derives the scale score.

3.4 Statistical Techniques

The measures appropriate to this work are measures of central tendency, which are used to discover a representative or typical value in the data set. These are:

- Mode - The most frequently occurring value. Multiple modes can exist in a data set
- Median - This is a positional value in the centre or middle of an ordered set. It is found by ordering the scores from lowest to

highest and if there are an odd number of scores selecting the middle score. If there are an even number of scores the middle of the two middle scores is selected.

- Mean - This is the arithmetic average for the scores in the data set. It is calculated using by summing all the scores and dividing by the number of scores in the data set.

Chapter 4 Method

The method adopted was to survey organisations and individuals to obtain information relevant to the research questions, analyse similar published surveys conducted elsewhere in the world and analyse existing estimating techniques.

The survey questionnaire was designed to cater for both individual and organisational aspects of interest in a combined manner. This was because:

- The research is aimed at both aspects of organisational and individual practice.
- The practitioners in the industry are fairly mobile moving from organisation to organisation and all experiences are relevant to this study.

4.1 Survey Procedure

A survey of software practitioners in Western Australia was conducted to ascertain the current state of practice in this geographical region.

1. A questionnaire instrument was developed addressing the research questions. This is contained in Attachment 1. **Note** questions not pertinent to this research are covered by the instrument for other research purposes.
2. The questionnaire was pre-tested with two academics that have had extensive (greater than twelve years) relevant industry experience.
3. Two pilot surveys were conducted with four Information Technology managers in Western Australia to tune the questionnaire instrument.

4. The survey was conducted in Western Australia using one hundred and two organisations selected at random from an existing database of two hundred and seventy eight organisations.
5. Analysis of the data to determine:
   - a classification of the project types;
   - methods and techniques used in estimate formulation;
   - accuracy of the above techniques;
   - commonality of elements estimated;
   - degree of validity of the estimates made;
   - conformity to best practice.

4.1.1 Subjects

The survey subjects were the people who performed estimates within their organisation. These were typically the Information Technology Managers and Project Managers involved in developing software intensive projects in a variety of organisations within Western Australia and responsible for providing the initial cost and effort estimates.

The organisations were selected by generating one hundred and two random numbers between one and two hundred and seventy eight and those organisations whose database key matched the generated number were used as the survey target. This method meets the random sampling criteria in that:

- every member in the population must have an equal opportunity of being chosen for the sample (equality) and

- the selection of one member is not affected by the selection of previous members (independence). Judd, Smith & Kidder (1991) p136.

The sample size was chosen by using the following assumptions:

Organisations

- Population size from which the sample will be drawn is 286
- Percentage of the population possessing the attribute of interest is 90%.
- Accuracy (precision) level .10 [low precision]
- Confidence level: .95
- Expected percent return rate is 30%

Individuals

- Population size from which the sample will be drawn is 2000.
- Percentage of the population possessing the attribute of interest is 90%.
- Accuracy (precision) level .10 [low precision]
- Confidence level. .95
- Expected percent return rate is 30%

Using

$$n = \frac{P((1-P)}{\frac{A^2}{Z^2} + \frac{P(1-P)}{N}}$$

leads to:

- for organisations a sample size of 103 organisations with 31 valid responses, and

- for individuals a sample size of 113 with 34 valid responses.

A covering letter was sent (see appendix 2) requesting that the organisation have its most experienced people complete the questionnaire. Reply paid envelopes were also included to encourage a good response rate.

As the initial response was not adequate follow up telephone calls were conducted to elicit a satisfactory response and obtain reasons for a lack of response. It was discovered on studying the reason for not responding that large number of companies out-sourcing their software development efforts and advised they did no estimating. An additional forty-four organisations were randomly selected and contacted in order to gain an adequate sample.

4.1.2 Instruments

The principle instrument was a survey questionnaire designed to elicit responses as to current practices at both an organisational and individual level. The questionnaire is described in more detail below. (See also Attachment 1).

4.1.3 Data Analysis

Classification of the estimating elements that the respondents use and their frequency and distribution.

4.1.4 Limitations

The selection of the organisations was from mailing list of potential customers of a consulting organisation. A bias may be present from the method(s) used to populate the database. The respondents are all volunteers with the corresponding attributes as discussed in chapter 3. It is assumed the respondents have some knowledge of estimating techniques. Non-respondents may have been reluctant to indicate a lack of knowledge or the use of good practice. Respondents may have given answers that reflected generally accepted good practice and not their actual practice. Several questions were posed in different manners to try to eliminate respondent bias.

The survey was limited to practitioners in the Western Australian region and the data can only be considered representative of that region.

4.1.5 Description of The Instruments Used

The collection of data was by way of questionnaire. See section 4.1.7 for a full description of the questionnaire.

4.1.6 Construction of the Questionnaire

The questionnaire was initially piloted by trialing with two experienced IT managers. This caused the wording of several questions to be altered in order to clarify their intent. Additional elements were also added such as the category "Sales People" to D.6. The questionnaire was then further tested with another two IT managers. This caused a major rewrite of the questionnaire to reduce its length by combining similar questions that related to new and maintenance projects. Also some questions were reformed to ensure they were clear in whether they related to the personal experiences and practices of the respondent or to the organisational practices.

4.1.7 The Questionnaire

The questionnaire consisted of the following five sections:

- Personal Details;
- Organisational Information;
- Software House or Information Technology Group Information;
- Software Estimation Practices;
- General Software Estimation Practices.

Each contained a number of questions relating to the subject topic of the section. The relevance of each question to the study is described below.

4.1.7.1 Part A: Personal Details

A.1 *Please indicate the number of years in the current organisation.*

This was designed to be used with questions in section E that relate to organisational estimation practices. The validity of the answers, especially where there are no formal estimating techniques (E.1), must be tempered against the history and length of time the individual had in an organisation.

A.2 *Indicate the total number of years experience you have in software development.*

Designed to see if there was any correlation between the number of techniques once used and discarded and the number of techniques currently used - E16 and E17. The data also adds weight to any conclusions drawn due the average respondent's length of time in the industry. It is postulated that a person who has been in the industry a number of years has had experience in this time. This is used only as an indicator of experience and not actual experience.

A.3 *Indicate the software project estimating techniques that you have used in your career. Please also rate their usefulness by circling a number. (1=useless;*

4=useful; 7=very useful).

The respondents could choose a number of techniques that they had used and indicate how useful they were on a seven point Lickert scale. An "Other" category was included to ensure all possibilities were accounted for. The techniques and their descriptions were

- **Analogy** (To compare past projects and extrapolate to new projects)
- **Expert Judgement** (To obtain expert opinions and apply to new projects)
- **Work Breakdown** (To define all the tasks in a project, estimate individually and then aggregate)
- **Lines of Code techniques** (To estimate the number of source instructions that will be delivered eg COCOMO)
- **Function Point Analysis** (Using Albrecht's techniques or a derivative)
- **Other** (please specify)

A.4 *Please rate your software project estimating skills. (1=poor; 4=average; 7=excellent).*

This was to determine the respondent's perception of their own estimating skills and to see the correlations with their number of years of experience (A.1) and the number of estimates conducted in the previous twelve months (A.5). This was also used to see the correlation between their perception and the adoption of what is considered good practice (E.1 to E.9)

A.5 *Approximately how many total project estimates have you made in the last twelve months?*

This question is used by several others for correlations and determines the amount of practice the respondent has had. It is also used to compare with E.15 to determine the percentage of estimates made by the individual in the organisation providing the response to A.1 is greater than twelve months.

4.1.7.2 Part B: Organisational Information.

B.1 *Please indicate if the organisation has a quality management system and its certification status.*

This question was designed to indicate whether or not the organisation had adopted what is generally accepted as good management practice and to see if this was reflected in their estimating practices (E.1 to E.9). There were a number of options given, which are outlined below. An "Other" category was included to ensure all possibilities were accounted for. The options covered the general International Standards Organisation's standards and the Australian standard AS 3563.1 that relates directly to the Design and Development of Software. The options given were:

- No Quality Management System
- Uncertified Quality Management system
- ISO 9001 certified
- ISO 9002 certified
- ISO 9003 certified

- AS 3563.1 certified
- Other. (please specify)

B.2 *Please insert the organisation's approximate total number of full time employees in Australia.*

This was used as an indicator of organisational size and to see if there were any differences in the practices between large and small organisations. This was combined with B.3 to form the total size of the organisation. The full and part time employees were split to ensure the response to the number of employees was unambiguous.

B.3 *Please insert the organisation's approximate total number of part time employees in Australia.*

See B.2 above.

B.4 *Please indicate your organisation's industry sector(s)?*

This question was designed to see if the practices differed between industry sectors. The sector categories have been derived from the Australian and New Zealand Standard Industrial Classification (ANZIC) coding system. (Australian Bureau of Statistics 1993).

The respondents were asked to classify their organisation at the top level. (The coding scheme has a second more detailed level). The categories are:

- agriculture
- mining
- manufacturing

- electricity/gas/water
- construction
- wholesale trade
- retail trade
- accommodation
- transport/storage
- communications
- finance/insurance
- property and business services (including software & consultancy houses)
- government/defence
- education
- health and community services
- cultural and recreational services
- personal and other services

4.1.7.3 Part C: Software House Office or Information Technology Group Information.

The heading wording was formed to ensure that the branch office of a software development organisation as well as a departmental software development group's information was gathered.

C.1 *Please indicate if the office or Information Technology Group has a Quality Management System.*

This question was designed to indicate whether or not the software house or department had adopted what is generally accepted as good management practice and to see if this was reflected in their estimating practices (E.1 to E.9). There were a number of options given, which are outlined below. An "Other" category was included to ensure all possibilities were accounted for. The options covered the general International Standards Organisation's standards and the Australian standard AS 3563.1 that relates directly to the Design and Development of Software. The options were:

- No Quality Management System
- Uncertified Quality Management system
- Within the scope of the organisation
- ISO 9001 certified
- AS 3563.1 certified
- Other. (please specify)

C.2 *Please insert the approximate number of people developing and maintaining software (including contractors).*

This was used as indicator of organisation size and to see if there were any differences in the practices between large and small organisations. It was also used to ascertain if there were different practices between organisations that used a large percentage of contractors.

C.3 *Please indicate the approximate number of contractors in C.2 above.*

Used to determine the percentage of contractors developing software. See C.2 above.

C.4 *Please indicate, by percentage, the types of software developed by the organisation.*

This question was used to determine if the software developed was for internal or external use. It was also used to see if they were any difference in practices and techniques used in organisations that developed for commercial purposes and those whose focus was in-house. The hypothesis being those organisations whose livelihood depends on their software products and services would have a more rigourous approach. The types were:

- applications for internal use
- applications for external clients
- commercial packages
- other - please specify

C.5 *Please indicate the approximate percentage of software projects undertaken by the organisation within the following categories.*

This question was designed to gather the percentage of project types developed in the organisation. This was to ascertain if there were any differences in practices and techniques used. It was also used to see if there was any correlation between the estimating accuracy (D.1) and the type of software developed. The categories used were derived in part from Pressman (1997) p15 and were:

- Business
- Engineering & Scientific

- System
- Real time/Embedded
- Other (please specify)

4.1.7.4 Part D: Software Estimation Practices.

This section related to general specific practices both at an organisational and personal level. The respondents were asked to differentiate between NEW and MAINTENANCE projects.

D.1 *Do your estimation practices differ from that of your colleagues? (1=Same; 4=somewhat different; 7=very different)*

This question had a seven point Lickert scale designed to see if there was any consistency in the organisation and if there was a correlation with B.1, C.1 and E.1 relating to formality of the organisation's management and estimating processes.

D.2 *Do you use different practices for different project categories? (1=Same; 4=somewhat different; 7=very different)*

Designed to see if specific techniques were used or related to specific project categories.

D.3 *Please state what percentage of development work is new development and maintenance.*

Designed to ascertain if certain estimating techniques were used in new developments versus maintenance work.

D.4 *Please indicate which software development methodology is used.*

The respondents were asked to differentiate between new and maintenance project practices. The method choices were:

- Informal
- Formal in-house
- Commercial

D.5 *Please indicate the software development lifecycle(s) used.*

The respondents were asked to differentiate between new and maintenance project practices.

- Waterfall or classical
- Prototyping
- Spiral
- Evolutionary
- RAD
- Object Oriented
- Other (please name)

D.6 *Please indicate who are* ***involved*** *in formulating the* ***initial*** *estimates for projects.*

The respondents were asked to differentiate between new and maintenance project practices. The choices were:

- Consultants
- IT Management

- Client Management
- Client Users
- Project Managers
- Analysts
- Programmers
- Specialist Estimating Staff
- Sales People
- Other (please specify)

D.7 *What lifecycle phases and other elements are typically included in the scope of your* ***initial nominal*** *project estimates? Please indicate the elements considered. Please also indicate your normal level of confidence in the estimate for the particular element.*

This question was designed to determine the elements that contributed to the estimate and whether the particular element was considered, when making an estimate, always, sometimes or never. The respondents were also asked to rate their degree of confidence for the particular element on a seven point Lickert scale. The table of elements was derived from Hope (1996). The elements listed were:

- Problem Definition & Feasibility Study
- Requirements Analysis
- Design
- Development (coding)

- Implementation
- Training
- Project Management & Administration
- Development Hardware and Software
- Operational Hardware & Software
- Environmental Changes
- Maintenance
- Other (please list)

D.8 *Please indicate the parameters which would cause the nominal estimates for projects to be modified. Indicate the parameters considered and the degree of impact they have on the nominal estimate.*

This question related to the parameters that would alter the nominal estimate and whether the particular element was considered, when making an estimate, always, sometimes or never. The respondents were also asked to rate the degree of impact for the particular parameter on a seven point Lickert scale. The cost driver parameters were derived from Hope (1996) and Boehm (1981). The parameter cost drivers were:

- People skills
- Knowledge of the application domain
- Complexity of the problem
- Algorithmic complexity
- Stability of the target platform

- User support
- Training
- Maintainability
- Performance requirements
- Usability
- Data base size
- Language to be used
- Reliability
- Project Risk
- Development environment
- Schedule constraints
- No of users
- Other (please list):

D.9 *For projects please indicate and rate your* ***normal*** *size, duration, effort and cost estimating practices.*

For each element (size, duration, effort and cost) the respondents were asked if the element was estimated and if it was they were requested to indicate the technique currently used by the organisation. They could select from:

- Analogy
- Expert Judgement
- Work Breakdown

- Algorithmic model or
- Other (specified)

For each technique they were then asked to indicate the:

- Frequency of use (1=never; 4=sometimes; 7=always)
- Ease of use (1=poor; 4=average; 7=excellent) and
- Accuracy (1=inaccurate; 4=accurate; 7=very accurate)

on a seven point Lickert scale.

This was designed to gain an insight into the practices of the respondent's organisation and to see if the elements were differentiated (ie did they make a distinction between effort and size) and the techniques used.

4.1.7.5 Part E: General Software Estimation Practices.

These questions related to total project estimates, the use of good practice and the revision mechanisms.

E.1 *What procedure does your organisation have for estimating?*

This was to determine the formality of the procedures adopted in the organisation. Used in correlation with the questions relating to quality management systems. The choices were:

- No procedure
- Informal procedure
- Documented procedure

E.2 *Do you use more than one technique to estimate a* **single** *project? (1=never; 4=sometimes; 7=always).*

Designed to see if the respondent has adopted good practice. Boehm (1981), (1997) and Pressman (1997) recommend the use of several techniques to overcome weaknesses of any single technique. A seven point Lickert scale was used.

E.3 *Do you keep records of project estimates? (1=never; 4=sometimes; 7=always).*

Designed to see if the respondent has adopted good practice. Boehm (1981), (1997) and Pressman (1997) recommend the use of several techniques to overcome weaknesses of any single technique. A seven point Lickert scale was used.

E.4 *Do you provide an indication of the probability of achievement for each estimate? (1=never; 4=sometimes; 7=always).*

Designed to see if the respondent has adopted good practice. Boehm (1981), (1997) and Pressman (1997) recommend the use of several techniques to overcome weaknesses of any single technique. A seven point Lickert scale was used.

E.5 *Do you produce a range of possible values for each estimate?*

(1=never; 4=sometimes; 7=always).

Designed to see if the respondent has adopted good practice. Boehm (1981), (1997) and Pressman (1997) recommend the use of several techniques to overcome weaknesses of any single technique. A seven point Lickert scale was used.

E.6 *Are the estimates peer reviewed? (1=never; 4=sometimes; 7=always).*

Designed to see if the respondent has adopted good practice. Boehm (1981), (1997) and Pressman (1997) recommend the use of several techniques to overcome weaknesses of any single technique. A seven point Lickert scale was used.

E.7 *How often are estimates revised?*

The choices were

- Never
- Only as specifications change
- At project review points
- Constantly under review
- Other (please note)

E.8 *How frequently do you compare your estimates with the actuals? (1=never; 4=sometimes; 7=always)*

Designed to see if the respondent had adopted the practice of a feed back mechanism and refined the estimates as the project progressed. A seven point Lickert scale was used.

E.9 *How frequently do you use a formal feedback mechanism to improve future estimates? (1=never; 4=sometimes; 7=always).*

Designed to see if the respondent had some historical record of estimates and if they were used to improve the estimates made in an organisation. A seven point Lickert scale was used.

E.10 *How important is software project estimating to your organisation? (1=not important; 4=important; 7=extremely important).*

Designed to gain opinion on the importance of the estimating to an organisation. A seven point Lickert scale was used.

E.11 *How satisfied are you with the estimation process within your organisation? (1=very dissatisfied; 4=okay; 7=very satisfied).*

Using a seven point Lickert scale respondent's opinions were sought. Used in correlations to see if there were relationships between technique usage and if the formality of the processes had any impact on respondents satisfaction levels.

E.12 *How accurate are the estimates made by the organisation. (1=inaccurate; 4=somewhat accurate; 7= very accurate)*

Designed to gain information as to the perception of the estimating accuracy within the software industry. Also to be correlated with A.4 that looks at the estimating skills the respondent believes they possess, and correlated with the responses to the accuracy component of D.9. A seven point Lickert scale was used.

E.13 *Why is estimating undertaken in the organisation?*

To access the reasons for estimating within the industry. The choices were:

- To provide firm quotations
- Organisational requirement
- To obtain project approval
- Assess project risk
- Budgeting

- Other. (please specify):

E.14 *In which phase of the project are you required to provide an **initial** estimate?*

(please specify):

This was an open question as there was a desire not to lead the respondents in any way. This was designed to support the hypothesis that estimates are required very early on in the projects when there are a large number of unknown factors. The responses have been categorised as:

- Quotation
- Feasibility study
- Requirements analysis
- Design

E.15 *Approximately how many total project estimates have been made in the **organisation** in the last year?*

Designed to elicit information as to the volume of estimates made within an organisation. The respondents were required to indicate estimates made for both new and maintenance projects.

E.16 ***Have you previously used** but **discarded** some estimating practices?*

This question had a yes/no option so if the respondent had not discarded any techniques they could easily move to the next question.

The techniques they could select from were:

- Analogy
- Expert Judgement

- Work Breakdown
- Algorithmic model or
- Other (specified)

They were asked to specify the software used if the technique was automated. They ' ere requested to specify the technique's name and its source from the following choices:

- Published in general literature
- Proprietary
- Developed in-house
- Developed by consultants

They were also asked to indicate the reasons why they had stopped using the technique from the following choices:

- Lack of training
- Found to be inaccurate
- Too difficult to use
- Too time consuming
- Inappropriate for your projects
- Other

E.17 *Please indicate **your current normal** estimating practices. Also indicate if the practice is **normally used** in your **organisation**.*

This question was designed to elicit what practices both the individual and the organisation used and also to see if there was any difference. A validity check on this question was conducted from D.1 which asked whether the individual's practices differed from their colleagues.

The techniques they could select from were:

- Analogy
- Expert Judgement
- Work Breakdown
- Algorithmic model or
- Other (specified)

They were asked to specify the software used if the technique was automated. They were requested to specify the technique's name and its source from the following choices:

- Published in general literature
- Proprietary
- Developed in-house
- Developed by consultants

They were also asked to indicate the reasons why the technique was used from the following choices:

- Training readily available
- Found to be accurate
- Easy to use
- Organisational standard
- Appropriate for the projects

- Other

E.18 *Please state your opinion on how you think the software intensive project estimating could be improved in your organisation?*

This was an open ended question designed to ascertain perceived and real weaknesses in organisational use of techniques and processes.

E.19 *Please state your opinion on how estimating techniques could be improved in the software industry.*

This was an open ended question to gain essentially expert opinion. It was also used to see if the responses from the practitioners in the Western Australia were similar to those gained by Park et al (1995).

E.20 *Please advise of any other aspect of the organisation's estimating practice that has not been covered by this questionnaire.*

This was an open ended meta question designed to give the respondent an opportunity to express any opinion on any subject not covered in the questionnaire.

4.1.8 Data Analysis

The data has been analysed using the statistical analysis software tool SPSS 7.0 using means, modes, medians, histograms and various other chart types. Correlations using Pearson's correlation coefficient were also determined as appropriate

4.2 Published Surveys

4.3 An analysis of published surveys of industrial organisations was conducted to determine:

- utilisation of existing techniques;
- perceived strengths and weaknesses of existing techniques;
- barriers to the use of existing techniques;
- desired attributes of an estimating technique.

4.4 Existing Technique Analysis

A detailed examination of existing techniques was conducted to determine:

- theoretical strengths and weaknesses;
- commonality of estimating elements and their attributes;
- explicit and implicit assumptions;
- estimating element coverage of the techniques;
- practical strengths and weaknesses.
- identification of weaknesses where major cost elements in a project were not catered for by the estimating technique.

4.5 Limitations

Estimating techniques examined were only those in the public domain. Other developers of estimating techniques that are proprietary and commercial, such as SLIM, do not publish algorithms and information relating to technique construction.

Chapter 5 Estimating Technique Survey Analysis

Estimating technique utilisation, which was obtained from three published surveys, one conducted in the USA another in New Zealand and the third in the Netherlands. Wydenbach & Paynter, (1995) also reported Heemstra & Kusters' (1989) results from a similar survey conducted in the Netherlands and Moores & Edwards (1992) results from the UK. (Hihn & Habib-agahi, 1991: Wydenbach & Paynter, 1995). The overall comparison table is reproduced in part in Table 5-1.

	Heemstra & Kusters (1989)	Moores & Edwards (1992)	Wydenbach & Paynter, (1995)
Survey Country	Netherlands	United Kingdom	New Zealand
Sample size	2659	115	515
No of respondents	597	54	236
Response rate	22%	47%	46%
Use of algorthmic models	14%	30%	26%

Table 5-1 NZ Survey Comparison

Hihn & Habib-agahi's (1991) survey was conducted by targeting those people in the organisation who estimated software costs. Their research was limited to the technical divisions of a single organisation, the Jet Propulsion Laboratory. There was no attempt to randomly select the sample population although it would appear they might have been attempting to question all involved in software estimating. Eighty-three people completed the questionnaire and also 48 completed some portion of a software size and effort estimation experiment. The respondents had a mean of 14.9 years of experience and showed a standard deviation of 7.6 years. They had conducted, on average, an estimate every eight months. The paper mentioned the extensive use of partitioning techniques which were grouped into three categories: function based, product based or algorithmic. The function based category was sub-divided into a high level functional breakdown and a low-level functional breakdown. The product based category was called the work breakdown structure category. Hihn and Habib-agahi (1991) described "the algorithmic category captures the computational process view of a software system". The results are in Table 5-2.

Partitioning Technique	Percentage
High level functional breakdown	53%
Low level functional breakdown	28%
Work breakdown structure	10%
Algorithmic	9%

Table 5-2 Partitioning Technique Percentage

The questionnaire contained four categories that were informal analogy, formal analogy, rules of thumb and models. The categorisation was not rigorous with overlaps and the data "reflects the authors' interpretation of what techniques were the dominant ones used". Hihn and Habib-agahi (1991) did not appear to consider the use of the various work breakdowns as a separate technique but as a first stage that other techniques were applied to. They recognised that multiple techniques would be used on a single project either in combination or as alternatives. They captured information relating to this multiple technique usage by identifying a primary and secondary approach for their respondents. The resultant data is in Table 5-3.

Estimation Technique	Respondent's Method	
	Primary %	Secondary %
Analogy, Informal	83	34
Analogy, Formal	4	0
Rules of Thumb	6	55
Models	7	11
Total	100	100

Table 5-3 Summary of Estimation Techniques

As can be seen the informal techniques are used by the greatest percentage of respondents in this survey and a small percentage use algorithmic models. Wydenbach & Paynter, (1995) conducted a survey where a questionnaire was mailed to 526 New Zealand organisations who, it was assumed, perform d some software development. The response rate was 46% and contained 213 use able responses. One question asked the frequency with which estimates were made. The results are reproduced in Table 5-4. It would seem that the respondents tend to make estimates more than some of the time.

Frequency	Number	Percentage
Never	4	2%
Some of the time	22	10%
Half of the time	8	4%
Most of the time	85	40%
Always	94	44%
Total	213	100%

Table 5-4 NZ Survey - Frequency of estimation

The question asking about the techniques used contained eight categories derived from Boehm (1981) that were:

- Expert judgement
- Reasoning by analogy

- Bottom up
- Automated or manual models (or both)
- Price to win
- Top down
- Available capacity
- Other

The respondents could choose more than one category. The results are in Table 5-5.

Estimation Methods	Percentage of Respondents
Expert judgement	86%
Reasoning by analogy	65%
Bottom up	51%
Automated or manual models (or both)	26%
Price to win	16%
Top down	13%
Available capacity	11%
Other	0%

Table 5-5 NZ Survey - Estimation Methods

The data resulting from a question regarding the importance of estimating indicates that whilst eighty percent of respondents consider the estimation process to be important and ninety eight percent make some form of estimate only 26% use a manual or automated algorithmic estimation model. The most common formal algorithmic model in use was found to be function point analysis with 20% of the total respondents who make an estimate using this technique. Table 5-6 below is a summary of data contained in these surveys. Where a method was not considered in a survey it has been marked not applicable (N/A).

Estimation Methods Wydenbach & Paynter	% of total respondents (209)	Estimation Methods Heemstra & Kusters	% of total respondents (369)	Estimation Methods Hihn & Habib-agahi	Respondents (83)	
					Primary %	Secondary %
Expert judgement	86%	Consult an expert	26%	Rules of thumb (expert)	6%	55%
N/A	-	Intuition	62%	Analogy, informal	83%	34%
Reasoning by analogy	65%	Analogy method	61%	Analogy, formal	4%	0%
Bottom-up	51%	N/A	-	N/A	-	-
Models	26%	Parametric models	14%	Models	7%	11%
Price-to-win	16%	Price-to-win	8%	N/A	-	-
Top-down	13%	N/A	-	N/A	-	-
Available capacity	11%	Capacity problem	21%	N/A	-	-
Other	0%	Other	9%	N/A	-	-

Table 5-6 Estimating Survey Summary

Heemstra & Kusters' (1989) data indicates that only fourteen percent use a formal model approach. This difference from the New Zealand survey (26%) was explained by Heemstra & Kusters' (1989) large percentage of the "other" category purports to contain non-commercial models.

It is interesting to note that in all surveys conducted above, the largest category was estimating by analogy.

Park, Goethert & Webb (1995) reported on a survey that was conducted in 1993 to assess the need for improvements in software cost estimating and as an input to the prioritisation of the work at the Software Engineering Institute (SEI) at Carnegie Mellon University. The survey was basic with only eight questions, one of which was contact information and another two related to obtaining sponsor involvement. They distributed the survey widely to groups affiliated with the SEI, at Software

Engineering conferences and those who had an obvious interest in software estimating such as the COCOMO users group. This survey can be said to be aimed at the software estimating experts. The results are based on the 249 responses received. The question that sought information on when and why estimates are used by industry sector is detailed in Table 5-7. The use categories would appear to overlap and respondents were instructed to select all categories that applied to them. Totals and percentages have been added to the original data.

Estimating Usage	Industry Sectors					
	Government/Military	Industry	Academic	Other	Total	Percentage
Project planning & scheduling	64	150	3	3	220	88%
Project staffing	43	122	0	3	168	67%
Estimates to complete	46	114	2	4	166	67%
Project preparation	36	115	2	3	156	63%
Replanning & rescheduling	37	101	2	2	142	57%
Project tracking	32	104	1	3	140	56%
Contract negotiation	31	80	1	4	116	47%
Proposal evaluation	43	64	0	2	109	44%
Resource levelling	20	60	1	2	83	33%
Concept exploration	25	54	2	3	84	34%
Design evaluation	25	52	2	2	81	33%
Bid/no-bid decision	13	63	2	2	80	32%
Other	17	11	2	1	31	12%

Table 5-7 SEI Survey - Estimate usage

As can be seen from the data in Table 5-7 the principal reason estimating is undertaken relates to project planning and management. The second reason relates to activities prior to the project commencing such as contract negotiation and proposal evaluation. The question of most interest in this research was "What improvements would be of most help?" This question did not have a structured reply and the authors grouped the responses according to the general areas they addressed and advised "... everyone sees a need to improve software estimating, but few see the same needs". The general area groupings used were size, models, databases, metrics and process. Unfortunately Park et al (1995) did not supply the total data, however, gave forty-nine examples of the responses. Of these, fourteen were concerned with the improvement of the sizing of a software project and thirty one advised a standard model and/or process with which to develop and record estimates would be of benefit. There were several who emphasised the need for historical data of actual results stored in databases. Park et al (1995) also sought suggestions on how organisations could improve their estimating. The resultant suggestions are in Table 5-8. An interesting observation from these suggestions is that they do not discuss building better algorithmic type estimating models. They discuss the need for a database containing historical data and also capturing and validating current project data. This would imply that the respondents find this information most useful in developing their own estimates. A database could assist by allowing reasoning by analogy, a listing of proposed and actual activities and for calibrating their models.

Organisation Sector	
Government & Military	**Industry**
• Maintain a comprehensive database of historical metrics and definitions • Make {cost, schedule, performance, metrics, measurement} a routine management tool • Management must require estimates for a useful purpose • Train future software cost estimators • Our organisation should develop a policy that dictates standard usage of estimating process, methods and tools. Due to a lack of policy, estimating is at the discretion of whoever is "in charge" of the proposal, project, program, etc.	• Capture data during and after the project • More formal analysis of project data collected. • Track actual hours worked vs. the 40 hour work week • Build a documented, realistic database of diverse results garnished from projects to be applied at the proposal/project start-up phases. • Perform follow-up data collection to assess the accuracy of the software estimates generated • Set up an effective metrics program (process metrics) • Project managers should define the metrics and data most useful to them for project planning and management.

Table 5-8 SEI Survey- Estimating Suggestions

Chapter 6 Analysis of Existing Estimating Models

6.1 Introduction

The formulation of any software metric must be defined with its intended use in mind. That is, without the clear specification of goals the metric is to achieve the measures will be of little practical benefit. This view is espoused by Fenton (1991) and Gilb (1988) who support Basili's Goal Question Metric approach to measurement (Basili & Rombach, 1988). Daskalantonakis (1992) provides practical experiences with this approach.

Whilst some work, such as Mukhopadhyay & Kekre (1992), has been published that addresses some of the issues involved with software estimating, few with the exception of Kitchenham, Pfleeger & Fenton (1995) have addressed the fundamental theoretical issues that form a necessary scientific basis for any technique. Matson, Barrett & Mellichamp (1994) provide an assessment method through the use of several statistical models that relate software development effort to software size in terms of function points. They are concerned with the empirical data upon which the models are based and the lack of attention to the appropriateness of the models. Jorgensen (1995) in addressing issues relating to the prediction of maintenance effort concludes, after the examination of several prediction models, "a formal prediction model should not replace the use of expert predictions". This would support Boehm's (1981) Wideband Delphi approach.

6.2 Function Point Analysis

There is uncertainty as to what function points are actually measuring. Albrecht's Function Point Analysis and Jones' Feature Point Analysis are assumed to either measure size or functionality as perceived by the user of the software product. The view held by the International Function Point Users Group, IFPUG (1994) is somewhat confusing as they discuss both "as a measure of the functional **size** of information systems" and a "measure of functionality that the user requests and receives". Albrecht (reported in Symons 1988) stated that the "measure isolates the intrinsic **size** of the system from environmental factors...".

However, function points are calculated from the sum of a number of different elements and therefore appear to be an attribute in their own right derived from an attribute relationship model as per Figure 3-1. As Kitchenham et al (1995) espouse, "the term function point does not seem appropriate; function points might be better renamed as functionality or user requirement size".

However, more elementary issues need to be addressed with function points. As described in chapter 2, Function Points are the sum of five elements derived from the number of inputs, outputs, inquiries, data and interface files. The input element is based on the number of data elements involved in each system input. If the number of data elements involved in all inputs were summed then this would be an acceptable measure of input data size. However, the function point model involves classifying each input as low, average or high, using an ordinal scale, according to the number of data elements and files accessed. The values derived are then mapped to numbers and summed. It would appear that the function point model is in violation of basic measurement theory in that you cannot sum ordinal scale measures. Abran and Robillard (1996) have addressed the validity of the mathematical transformations in counting function points. Firstly they defined the counting process for the data elements as follows.

Step F1 - Obtain relevant documentation and identify related groups of data in order to produce a list of logical files.

Step F2 - Classify the logical groups of data in order to produce lists of internal logical files and external interface files.

Step F3 - Count the data and record elements in order to produce a data element (DET) count and a record element (RET) count.

Step F4 - Apply the data algorithm using the DETs, RETs, data matrix table and associated weights (see Table2-1). This produces a list of points for all logical files.

Step F5 - Add all the points from Step F4 to produce the unadjusted data count.

The scale transformations for Step F4 has been further sub-divided into four sub-steps in order to identify all the measurement transformations that occur. The results of the transformations and scale analysis are contained in Table 6-1.
(See chapter 2 for a full discussion on counting function points and also see chapter 3 for a discussion on the various scale types and the permissible mathematical operations).

Step	Objects	Operation	Scale from	Scale to	Math validity	Implicit transformation
F3	Data	Count	Absolute	Absolute	Yes	No
	Record	Count	Absolute	Absolute	Yes	No
F4a	Data	Identify range	Absolute	Ordinal	Yes	Yes & loss of information
	Record	Identify range	Absolute	Ordinal	Yes	Yes & loss of information
F4b	Function of ranges of (data, record)	Position in matrix	Ordinal	Nominal	Yes	Yes & loss of information
F4c	Function of position in matrix	Name & order	Nominal	Ordinal	**No**	Yes & addition of information
F4d	Function of perceived values	Assign weights	Ordinal	Ratio	**No**	Yes & addition of information
F5a	Weights of internal files	Add	Ratio	Ratio	Yes	No
	Weights of external files	Add	Ratio	Ratio	Yes	No
F5b	Weights: internal + external	Add	Ratio	Ratio	Yes	No

Table 6-1 FPA Scale Transformation

As can be seen in moving from Step F3 to Step F4b the measurement scales go from the absolute scale to the ordinal scale and then to the nominal scale. These transformations are mathematically valid however involve a loss of measurement information and a reduction in the valid operations allowed. Step F4c and Step F4d move in the opposite direction. The transformation noted for data also apply to the transaction and VAF processes. All the transformations have been made possible by the assigning of weights to transform the five different types of objects that make up a function point count (internal files, external interface files, inputs, outputs and inquiries) into a single unspecified object of a different type. Therefore as Abran and Robillard (1996) pp 899 say:

"The end results (unadjusted and adjusted function points) become, therefore, very difficult to interpret: there are so many dimensions involved and so many uses of different types of scales that the end measure, which might look rather simple and reasonable, is, in fact, a pot pourri that might not have correct mathematical meaning."

Also the counting rules mean that the smallest system has a value of three because the simplest enquiry on a single file with one attribute maintained by another system takes this value. This implies that the values are discontinuous and there is no unit value. This is another violation of the measurement framework. The classification of system component types (input, logical files etc) into low, average and high is also an oversimplification. For example a logical file that contains one data element is given a count of seven whereas a logical file containing a hundred data elements is given a count of fifteen which is only just over twice the value. These arguments are also applicable to Feature Points.

Albrecht's Function Points have also been criticised by Symons (1988) on a number of grounds. These being:

- It is difficult to define the basic counts objectively.
- The low, average and high classification is over simplified.
- The choice of weights for the initial classification and calculation of the technical complexity factor was determined subjectively and based on experiences at IBM.

- Internal complexity is treated twice, during the initial classification and during the calculation of the technical complexity factor.
- The effect on function point counts of comparing a group of independent systems linked by interfaces and a single fully integrated system is counter intuitive.

There are also problems with the value adjustment factors in several ways. Jeffrey, Low & Barnes (1993) have shown that the complexity adjustments do not improve effort predictions and there were no significant differences between unadjusted and adjusted function points as effort predictors. Kitchenham & Kansala (1993) have reported similar results.

Fenton (1994) is of the opinion that using the VAF adjustment, for a model that measures system functionality, is "analogous to redefining measures of height of people in such a way that the measures correlate more closely with intelligence". Other concerns with VAFs is that they are open to interpretation and it is easy to see overlap. See Table 6-1 for details of overlap.

Therefore the use of VAFs are subjective and depends on interpretation as to what the person conducting the count perceives as being in each category. VAFs were formulated in 1984 and as such are not wholly relevant to modern software products and development environments. For instance, the graphical capabilities required and the provision of inquiries as defaults in fourth generation languages are not easily accounted for.

VAF		VAF Overlap
1.	Data communications	6, 8, 2
2.	Distributed data processing	1
3.	Performance	6, 8
4.	Heavily used configuration	
5.	Transaction rate	
6.	Online data entry	1, 3, 8
7.	End-user efficiency	6, 8
8.	Online update	1, 3, 6, 7, 14
9.	Complex processing	
10.	Reusability	
11.	Installation ease	
12.	Operational ease	
13.	Multiple sites	
14.	Facilitate change	

Table 6-2 VAF Overlap

Symons (1988) reports the function point complexity weights chosen "was determined by debate and trial" and suggests a more objective approach should be taken. One of the more important modifiers to most other estimating techniques are aspects of the quality of the software product, most of the quality attributes are missing from the function point model.

The application of the model will always give a linear result which is counter-intuitive in that the amount of work increases geometrically as the size of the project increases ie large projects take a significant amount of more work than small ones. The applicable scope of a software project covered by function points is undefined. This would appear to be a major omission as one of the stated aims of IFPUG (1994) is to provide a normalisation factor for software comparison. The least the Function Point models should do is outline the lifecycle phases and major activities that are part of the "size".

Mark II function points takes a different approach in that the function points are derived from the inputs, outputs and entities for each business transaction. The transaction input size is the sum of the data elements that are input into the system; the transaction output size is the sum of the data elements that are output from the system; the transaction data processing size is the sum of the number of entities referenced when the transaction is processed. These values are summed for each transaction and therefore represent three different size attribute elements that are input into the system. The model requires that the attribute values be weighed and summed. The weights are different for each attribute and represent the development effort involved. This violates the measurement framework if we regard Mark II function points as a size or functionality measure, however, it could be considered to be an effort measure as the weights are derived from the number of man-hours involved in delivering each component.

It must be concluded that there are major problems associated with the meaning and construction of function point measures. It is interesting to note that there is little work published on the validity of the measures as to their predictive capability.

6.3 COCOMO & Lines of Code Measures

The original COCOMO model depends on estimates of size in terms of KDSI (thousands of delivered source instructions) for its major input which is not really measurable until the software product has been implemented. As such this measure is subjective although estimates should become more accurate as the project progresses. However at the commencement of a project it would appear that difficult problems of estimating or predicting effort, schedule and cost are replaced with the equally difficult problem of estimating size. Also the COCOMO models require that the modes of development (organic, semi-detached or embedded) be determined and in the Intermediate and Advanced models fifteen cost drivers must also be rated. These are based on subjective opinion and therefore the objectivity of the inputs to the COCOMO models is questionable.

The use of KDSI has other problems that are as follows.

- As Jones (1991) states there is no industry standard definition for a line of code (LOC).
- Some languages such as Pascal and Ada allow many logical statements per physical line whereas other languages such as COBOL have physical line requirements.
- The types of lines that are counted need to be defined as most procedural languages include four different kinds of source statements executable lines, data definitions, comments and blank lines. Data definitions can also cause problems as *n* variables can be declared in one statement or *n* statements for the same logical outcome.
- The concept of a LOC is not represented in some fourth generation languages such as Oracle Forms. These languages also tend to use third generation type languages in part, thereby compounding the problem.

COCOMO 2.0 recognises the above problems and uses several different models in an attempt to overcome the issues. However the inputs to these models are equally as suspect. For instance the COCOMO 2.0 stage 2 model uses function points or source lines of code (SLOC) as the input size measure and hence has the problems as discussed above.

Another problem associated with lines of code is that they are not all equal. Even using a detailed standard to count SLOC different lines will have a different degree of complexity and hence it is difficult to equate one line with another.

6.4 Conclusion

Function points do not relate to any lifecycle model or any set of activities. Therefore in addition to the problems mentioned above it is difficult to know what activities can be included when determining productivity and costing factors. That is, is it allowable to include such elements as the effort to produce systems manuals, the cost of development tools etc in the production of the system under investigation. COCOMO has a model on which it is based and only covers the software lifecycle from requirements to implementation for those activities in the work breakdown structure nominated. However, it has all the problems associated with estimating lines of code. COCOMO 2.0 addresses some of the issues however uses function points, object points or lines of code as inputs to the various models.

It should be noted that no published material was found relating to experiences with the Wide-Band Delphi method.

Chapter 7 Analysis of Current Practice: WA Survey

7.1 Overview

A survey was conducted to ascertain the current software project estimating practice within Western Australia. A questionnaire (appendix 1) was sent to 108 organisations selected at random from a commercial mailing list database containing a total of 384 organisations. The company, Spiral Technology Pty Ltd, who supplied the mailing list from its customer database, provides software engineering consultancy services to Western Australian organisations. It was considered these organisations would have some involvement or interest in the software development processes and hence software project estimating. The mailing list has been actively managed and updated since 1993.

A letter (see appendix 2) accompanied the questionnaire and requested that the most knowledgeable estimating people complete it. The objective of the survey was to obtain opinion from people considered expert in their organisation and who actively practiced estimating. The initial response to the questionnaire was twenty-four which was considered too low to perform a valid analysis and so a further thirty-six organisations were selected at random and questionnaires mailed to them. All companies mailed were followed up with several telephone calls to maximise the response rate and to determine the reasons for not responding. The total number of companies responding was thirty-seven. However, in some companies several people completed the survey and hence a total of forty-five completed forms were received. One response was unusable and excluded, hence forty-four have been included in these results. The response results are in Table 7-1 Survey Responses. An assumption was that even if the organisation did not develop their own software, outsourced its development or purchased packages it would be estimating to assure the contracts let were reasonable or it was prudent to purchase packages. As can be seen from Table 7-1 this assumption was proved false with a considerable number of organisations, forty-eight, considering the survey not to apply to them.

	Number
Returned	37
Not Applicable • No software development undertaken. • All software development outsourced. • Only use commercial packages.	48
Develop software however too busy to respond	9
Do not respond to surveys	5
Unable to obtain any response	43
Misc - A company was duplicated in the database and another no longer exists.	2
Total	144

Table 7-1 Survey Responses

7.2 Organisation Demographic Findings

The responses were from a broad range of industry sectors as can be seen in Organisations by Industry Sector Table 7-2. Figure 7-1 a "Graph Organisations by Industry Sector" also shows the distribution of the responses clearly. The responses are what were expected from Western Australian industry in that the organisations reflect what is generally regarded as the make-up of the organisations in Western Australia. That is, there is a predominance of government/defence and mining organisations represented. It should be noted the sector " property & business services" contains software development and consultancy organisations and hence this category is well represented. It is assumed the survey contains data that reflects the organisations within Western Australia although comparison data from the Australian Bureau of Statistics or the state government was not available.

Industry Sector	Frequency	Percent
mining	6	13.6
manufacturing	3	6.8
electricity/gas/water	2	4.5
construction	1	2.3
retail trade	1	2.3
finance/insurance	5	11.4
property & business services	11	25.0
government/defence	11	25.0
education	2	4.5
health & community services	1	2.3
Total	43	97.7
Missing	1	2.3
Total	**44**	**100.0**

Organisations by Industry Sector Table 7-2.

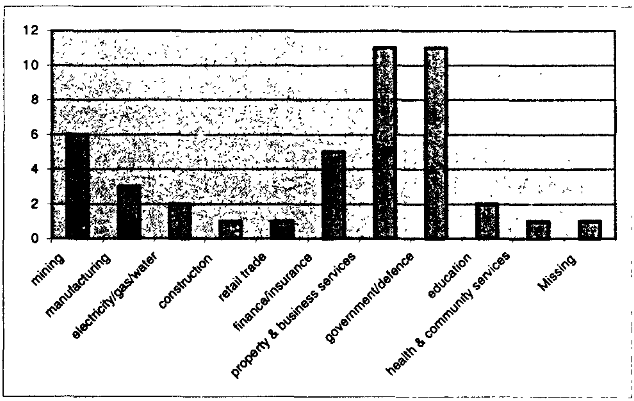

Figure 7-1 Organisations by industry sector

The organisations range in size from having four to twenty three thousand full time employees. As can be seen from Table 7-3 the spread across the broad range is even and consistent.

No of Full Time Employees	Frequency	Percent	Cumulative Percent
4.00	1	2.3	2.3
10.00	1	2.3	4.5
12.00	1	2.3	6.8
30.00	1	2.3	9.1
40.00	2	4.5	13.6
60.00	1	2.3	15.9
70.00	1	2.3	18.2
75.00	1	2.3	20.5
130.00	1	2.3	22.7
150.00	1	2.3	25.0
155.00	2	4.5	29.5
200.00	3	6.8	36.4
250.00	1	2.3	38.6
330.00	1	2.3	40.9
350.00	1	2.3	43.2
400.00	1	2.3	45.5
450.00	2	4.5	50.0
550.00	1	2.3	52.3
581.00	2	4.5	56.8
700.00	1	2.3	59.1
760.00	1	2.3	61.4
800.00	4	9.1	70.5
1000.00	2	4.5	75.0
1250.00	1	2.3	77.3
1500.00	1	2.3	79.5
1600.00	1	2.3	81.8
1650.00	1	2.3	84.1
2000.00	1	2.3	86.4
2600.00	1	2.3	88.6
5500.00	2	4.5	93.2
5847.00	1	2.3	95.5
10000.00	1	2.3	97.7
23000.00	1	2.3	100.0

Table 7-3 Number of full time employees

The result is what is expected from the Western Australian industry with fifty percent having fewer than four hundred and fifty full time staff. These figures add weight to the claim that this study is representative and hence the conclusions drawn have validity.

Another aspect that typifies the Western Australian industry is the small size of the development groups. The graph in Figure 7-2 demonstrates the small nature of the development groups.

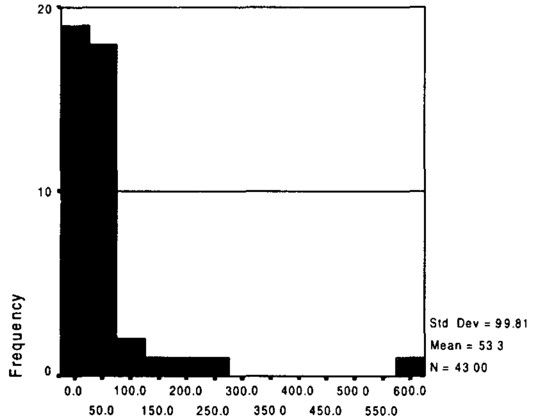

Figure 7-2 Number of people in software development groups

It should be noted that the overall result is skewed by the response indicating six hundred people are involved in the development and maintenance of software within their organisation. Table 7-4 summarises the situation and gives a mean of 53.3 however the mode is only 30 which tends to indicate the relatively small size, by world standards, of the software development organisations in this study. This size is however indicative of the organisations within Western Australia

	Valid	Missing	Mean	Median	Mode	Sum
No of s/ware people	43	1	53.26	30.00	30	2290

Table 7-4 Number of development people statistics

The number of software development people represented in the organisations that responded is two thousand two hundred and ninety. This represents a considerable percentage of the software developers in Western Australia. Especially so when one considers the membership of the Western Australian branch of the Australian Computer Society is one thousand four hundred.

The type of development work undertaken is for internal use, external clients and commercial packages. The majority of the work is mainly for internal use within the organisation with the mean being 65.1%. This can be seen in Table 7-5 that summarises the types of software developed.

	Valid	Missing	Mean	Median	Mode
% of internal use apps	41	3	65.0976	95.0000	100.00
% of apps for external clients	36	8	33.7222	10.0000	.00
% of commercial packages	29	15	10.4828	.0000	.00
Other	1	43	5.0000		5.00

Table 7-5 Types of software developed

The graph in Figure 7-3 also clearly shows that whilst the majority of respondents developed applications for internal use a considerable number developed applications for specific clients (thirty six) or as commercial packages (twenty nine).

The graphs in Figures 7-4 and 7-5 further illustrate this point.

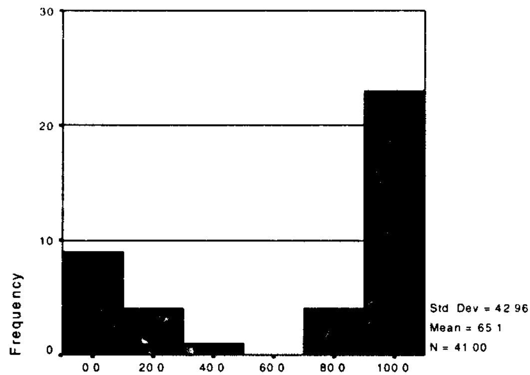

Figure 7-3 Percentage of internal development work undertaken

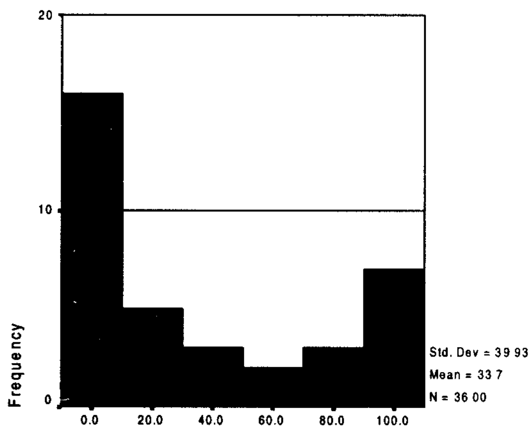

Figure 7-4 Percentage of external development work undertaken

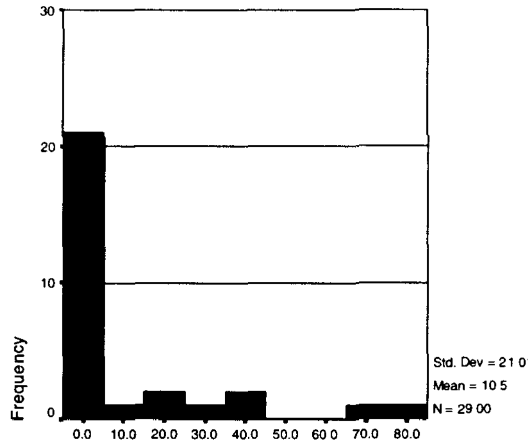

Figure 7-5 Percentage of commercial packages developed

It should be noted that no organisation solely developed commercial packages although thirty-one percent of the respondents undertook some development activity in this area.

Table 7-6 shows the number of organisations developing software projects in various categories and the percentage of work undertaken in each category on average.

	Valid	Missing	Mean	Median
% of business projects	44	0	77.1364	90.0000
% of engineering & scientific projects	34	10	8.7941	2.5000
% of system projects	34	10	8.9706	5.0000
% of real time/embedded projects	30	14	7.4000	.0000
% of other projects	29	15	6.1034	.0000

Table 7-6 Percentage of project categories

It shows that a broad range of project types is undertaken although the majority of work is classified as business projects. The graphs 7 6, 7-7, 7-8, 7-9, 7-10 illustrate the range and number of organisations in each category.

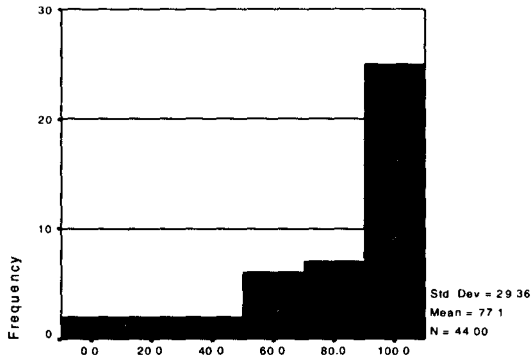

Figure 7-6 Percentage of business projects

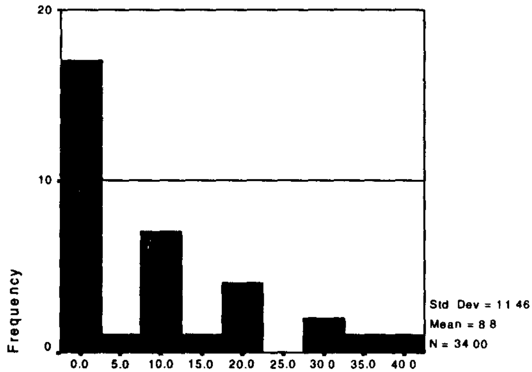

Figure 7-7 Percentage of Engineering & scientific

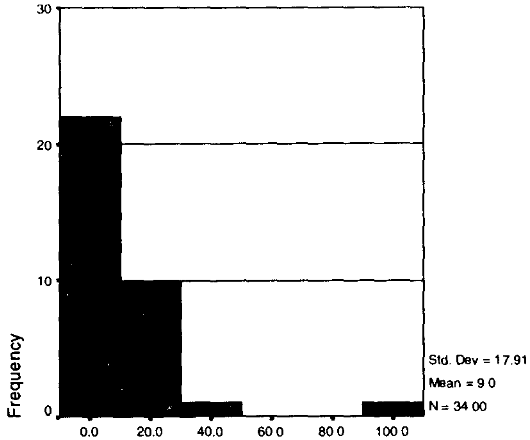

Figure 7-8 Percentage of system projects

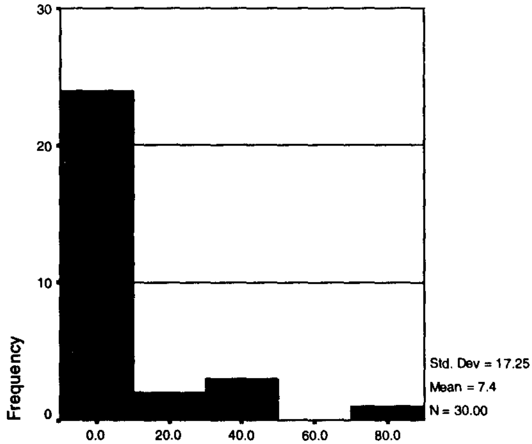

Figure 7-9 Percentage of real time/embedded projects

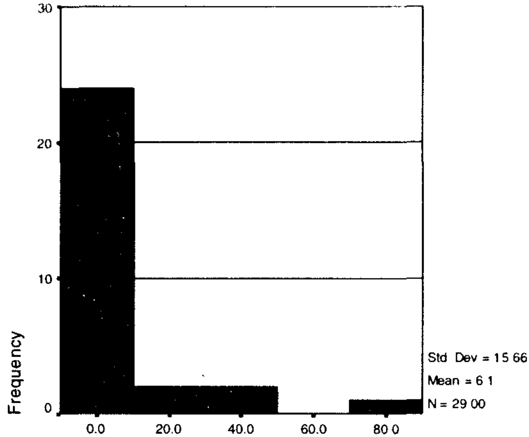

Figure 7-10 Percentage of other projects

As can be seen from the tables and graphs above the majority of the projects were able to be associated with the specified categories.

Therefore, it can be said the respondent organisations to the survey generally represent the Western Australian organisations developing software in that:

- A range of different sized organisations responded although they tended to be of a small nature;
- Different industry sectors were represented with mining and government being prominent.
- The application types developed were for internal use, built for specific external clients and commercial packages were developed.

The categories, derived from Pressman (1997), covered the range of projects that are generally considered undertaken within the industry.

7.3 Respondents Demographics

The total years of software development experience is detailed in Table 7-7 Total Years of Software Development Experience.

Years of Experience	Frequency	Percent
4	1	2.27
7	1	2.27
8	2	4.55
10	4	9.09
11	2	4.55
12	4	9.09
14	3	6.82
15	7	15.91
16	1	2.27
17	2	4.55
18	3	6.82
19	3	6.82
20	4	9.09
21	1	2.27
22	1	2.27
24	1	2.27
25	1	2.27
27	1	2.27
36	1	2.27
Total	43	97.73
Missing	1	2.27
Total	44	100.00

Table 7-7 Total Years of Software Development Experience

As can be seen from this table and the graph in Figure 7-11 Total Years of Software Development Experience the aim to have experienced people complete the survey has been achieved. Of the forty-three valid responses only four people, representing

4.55%, have less than ten years experience in the industry. The mean being nearly sixteen years also demonstrates the depth of experience of the respondents.

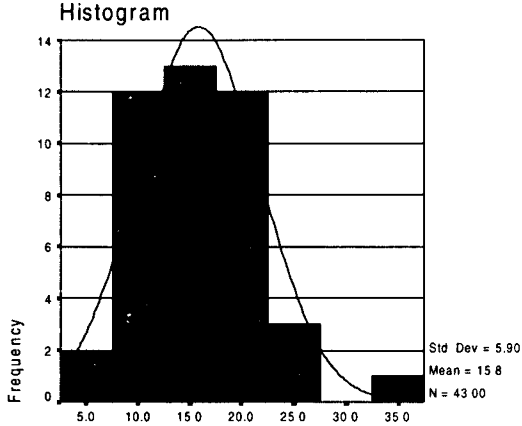

Figure 7-11 Total years of software development experience

Therefore, the results of this survey can be said to originate from people who have been in the industry for a considerable time. Hence these people should be experienced in software development and have had exposure to a number of estimating techniques in their careers.

This degree of experience is further supported in that the majority of respondents have worked in more that one organisation as the number of years worked in the current organisation has a mean of 8.5 years. Figure 7-12 clearly shows this and also shows a bunching towards the lower end of the scale which indicates the majority of people spend less time than the mean in an organisation.

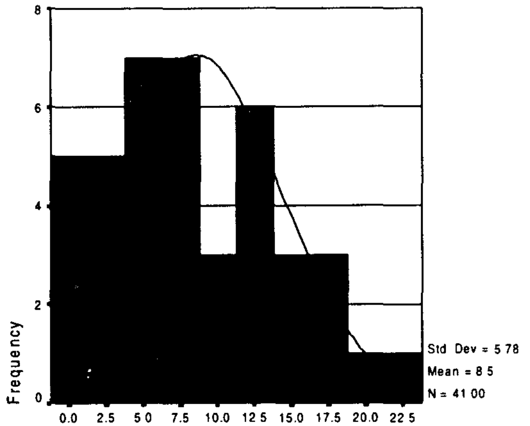

Figure 7-12 Years in current organisation

7.4 Estimation Practices

A number of questions were asked in relation to the respondents' estimating experience with different techniques and their view of how useful the techniques were.

The categories used were derived from previously conducted surveys (see section five) and what has been generally published in the literature.

The rating scale had seven points with one being useless, four being useful and seven being very useful. Table 7-8 summarises the data.

	Valid	Missing	Mean	Median	Mode
Use of analogy	40	4	5.25	6.00	6
Use of expert judgement	40	4	5.18	5.00	6
Use of work breakdown	42	2	5.45	6.00	6
Use of LOC	22	22	2.59	2.00	2
Use of FPA	23	21	3.91	4.00	4
Use of other	4	40	4.00		4

Table 7-8 Summary of estimating practice use and rating

Table 7-9 details the responses regarding the use of analogy and Figure 7-13 illustrates the data.

	Rating	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	1	2.3	2.5	2.5
	3	2	4.5	5.0	7.5
	4	6	13.6	15.0	22.5
	5	9	20.5	22.5	45.0
	6	20	45.5	50	95.5
	7	2	4,5	5.0	100.0
Total		40	90.9	100.0	
System Missing		4	9.1		
Total		44	100.0		

Table 7-9 Use and rating of analogy

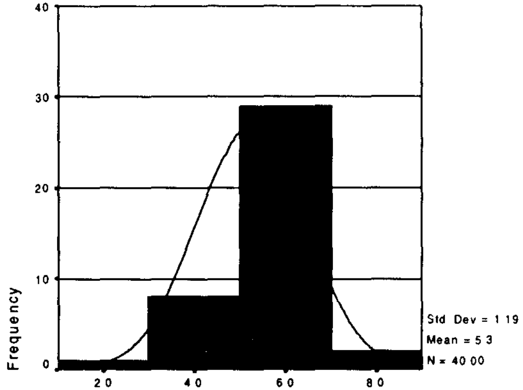

Figure 7-13 Use and rating of analogy

The data shows 91% of the total respondents have used the analogy technique and in general have found it more than useful with the mean being 5.3.

Table 7-10 details the responses regarding the use of expert judgement and figure 7-14 illustrates the data.

	Rating	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	1	2.3	2.5	2.5
	3	1	2.3	2.5	5.0
	4	9	20.5	22.5	27.5
	5	11	25.0	27.5	55.0
	6	14	31.8	35.0	90.0
	7	4	9.1	10.0	100.0
Total		40	90.9	100.0	
System Missing		4	9.1		
Total		44	100.0		

Table 7-10 Use & rating of expert judgement

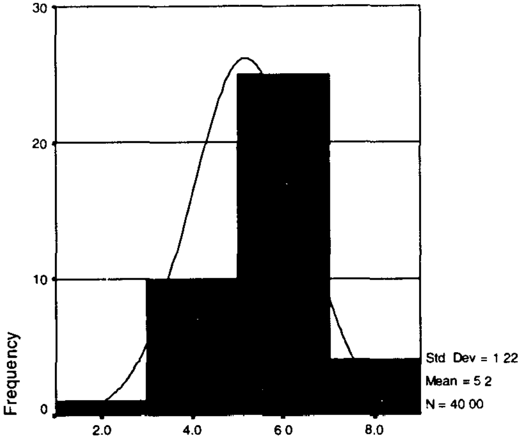

Figure 7-14 Use and rating of expert judgement

Ninety-one percent of the total respondents have used the expert judgement technique and in general have found it more than useful with the mean being 5.2. Table 7-11 details the responses regarding the use of work breakdown and figure 7-15 illustrates the data.

	Rating	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	1	2.3	2.4	2.4
	3	3	6.8	7.1	9.5
	4	5	11.4	11.9	21.4
	5	8	18.2	19.0	40.5
	6	16	36.4	38.1	78.6
	7	9	20.5	21.4	100.0
Total		42	95.5	100.0	
System Missing		2	4.5		
Total		44	100.0		

Table 7-11 Use and rating of work breakdown

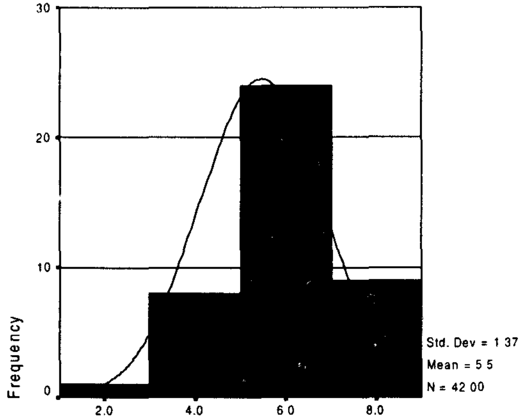

Figure 7-15 Use and rating of work breakdown

Ninety-five point five percent of the total respondents have used the work breakdown technique and in general have found it more than useful with the mean being 5.5.

Table 7-12 details the responses regarding the use of LOC (Lines of Code) and figure 7-16 illustrates the data.

	Rating	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	6	13.6	27.3	27.3
	2	8	18.2	36.4	63.6
	3	4	9.1	18.2	81.8
	5	2	4.5	9.1	90.9
	6	1	2.3	4.5	95.5
	7	1	2.3	4.5	100.0
Total		22	50.0	100.0	
System Missing		22	50.0		
Total		44	100.0		

Table 7-12 Use and rating of LOC

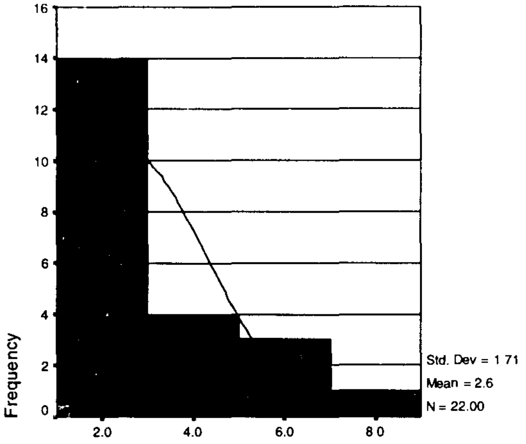

Figure 7-16 Use and rating of LOC

Only fifty percent of the total respondents have used Lines of Code (LOC) techniques and in general have found them less than useful with the mean being 2.6. Sixty three point six percent found the techniques to be useless or next to useless and only nine percent found them to be better than useful.

	Rating	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	2	4.5	8.7	8.7
	2	3	6.8	13.0	21.7
	3	4	9.1	17.4	39.1
	4	5	11.4	21.7	60.9
	5	5	11.4	21.7	82.6
	6	3	6.8	13.0	95.7
	7	1	2.3	4.3	100.0
Total		23	52.3	100.0	
System Missing		21	47.7		
Total		44	100.0		

Table 7-13 Use and rating of Function Point Analysis

Table 7-13 details the responses regarding the use of Function Point Analysis and Figure 7-17 illustrates the data.

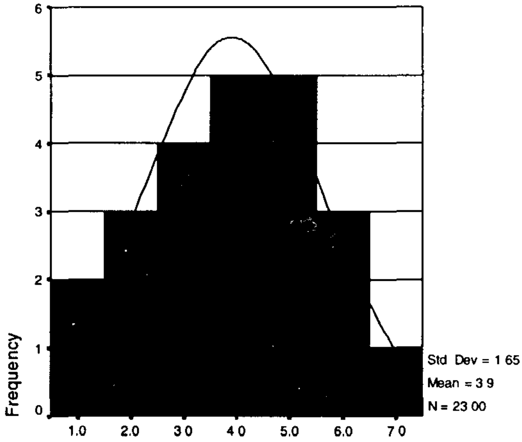

Figure 7-17 Use and rating of Function Point Analysis

Fifty two point three percent of the total respondents have used the Function Point Analysis (FPA) technique and in general have found it to be useful albeit marginally so. The data indicates that 39.1% found FPA to be less than useful and 39% found it to more than useful with the mean being 3.9.

Table 7-14 details the responses regarding the use of "Other" techniques. As can be seen this only represents four of the respondents or nine point one percent all of who indicated they found the technique they used to be useful.

	Rating	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	4	4	9.1	100.0	100.0
	Total	4	9.1	100.0	
System Missing		40	90.9		
Total		44	100.0		

Table 7-14 Use and rating of Other Techniques

Other techniques were specified by four respondents and these were:

- Feature Points (the Capers Jones derivative of Function Point Analysis);
- Joint Estimating Workshops - this is assumed to be an implementation of Boehm's Wide Band Delphi or similar;
- Guestimate;
- Complexity Matrix.

The survey appeared to capture the estimating techniques that are and have been prevalent in the software development industry and published in the general literature.

The respondents were asked to rate their own estimating skills with one representing poor, four representing average and seven excellent. The results are shown in Table 7-15 and Figure 7-18. The response is as expected with only a small percentage, 6.8% rating themselves below average. The people who rated themselves below average only did so by one point. None of the respondents rated themselves as having poor estimating skills. This result was expected as the survey was aimed at experienced and expert estimators in the organisations. This rating of course does not imply the respondents give accurate estimates only that they rate their skills as better than average.

	Rating	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	3	3	6.8	6.8	6.8
	4	6	13.6	13.6	20.5
	5	24	54.5	54.5	75.0
	6	10	22.7	22.7	97.7
	7	1	2.3	2.3	100.0
Total		44	100.0	100.0	

Table 7-15 Rating of estimating skills

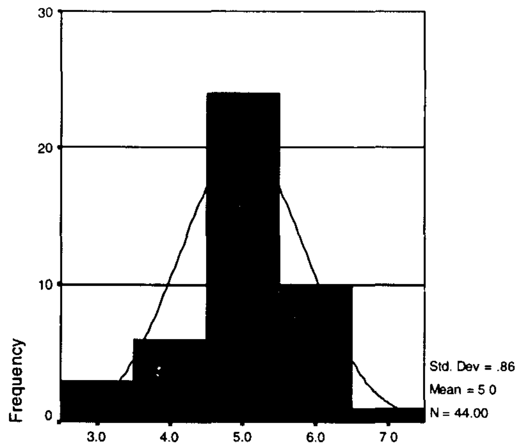

Figure 7-18 Rating of estimating skills

It is also interesting to note that whilst the majority of respondents (thirty five) rated their skills above average only one person gave themselves an excellent rating. This result was expected as it was assumed the people specialising in estimating in their organisation would have above average estimating skills and it would be the principal reason they were employed to perform this work.

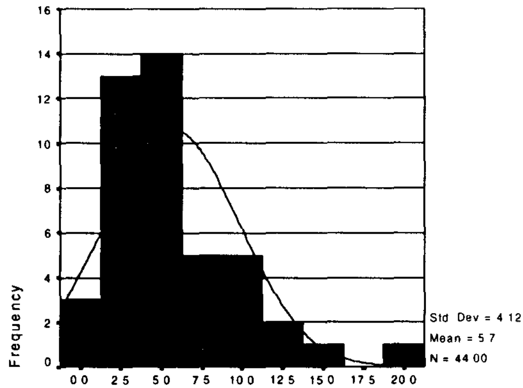

Figure 7-19 Estimates made in last 12 months

The total estimates from all respondents were two hundred and forty nine. This would indicate that on average the respondents would produce an estimate about every two months. This adds further weight to the claim the respondents have a high level of estimating expertise and practice.

The response to the question "Do your estimating practices differ from that of your colleagues?" where a seven point scale was used with 1 = same, 4 = somewhat different and seven = very different is detailed in Table 7-17 and Figure 7-20.

As can be seen the practices reported tended to similar with only 16.7% indicating their practices were somewhat different to very different from their colleagues. Only 20.5% indicated their practices were the same therefore it must be inferred there are slight differences in how the estimating process is undertaken between people in the same organisation. Only 6.8% advised their practices were more than somewhat different (a score greater than five) to their colleagues.

	Rating	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	9	20.5	21.4	21.4
	2	13	29.5	31.0	52.4
	3	13	29.5	31.0	83.3
	4	4	9.1	9.5	92.9
	5	2	4.5	4.8	97.6
	6	1	2.3	2.4	100.0
	Total	42	95.5	100.0	
System Missing		2	4.5		
Total		44	100.0		

Table 7-17 Similarity to colleagues' practices

Figure 7-20 shows the frequency of the responses clustered towards the end of the scale indicating the sameness of practice.

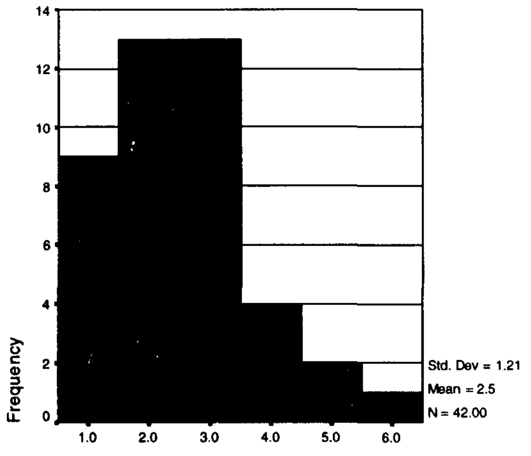

Figure 7-20 Similarity to colleagues' practices

Information was also sought regarding the estimating practices for different project categories. This is detailed in Table 7-18 and Figure 7-21.

	Rating	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	6	13.6	14.3	14.3
	2	14	31.8	33.3	47.6
	3	6	13.6	14.3	61.9
	4	10	22.7	23.8	85.7
	5	4	9.1	9.5	95.2
	6	1	2.3	2.4	97.6
	7	1	2.3	2.4	100.0
Total		42	95.5	100.0	
System Missing		2	4.5		
Total		44	100.0		

Table 7-18 Practices for different project categories

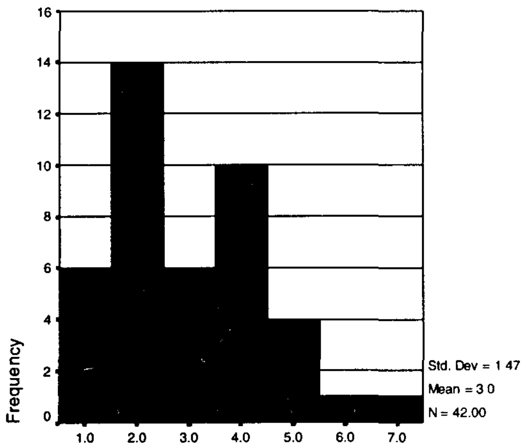

Figure 7-21 Practices for different project categories

14.3% advised they used the same practices for all projects and 38.1% indicated they used "somewhat different" to "very different" practices. This would tend to indicate that estimating practices were generally modified to some extent dependant on the project category.

7.5 Other Aspects of Estimating

It was intended to see if there was any differentiation between the estimating practices with new and maintenance projects. Unfortunately, the questions asking if the respondents' practices differed between new and maintenance projects were absent from the questionnaire. However, some interesting data was gathered and is in Table 7-19 and Figures 7-22, 23. This data shows that approximately 60% of the work is for new projects and forty percent is for maintenance projects. These figures are somewhat contrary to accepted wisdom, that is widely published in the literature, that typically puts new development at forty percent and maintenance at sixty percent or greater. For instance Takang and Grubb (1996) p13-15 report several refereed studies that are in line with the ratio quoted above. The data shows that 75.6% of the respondents had less than 70% of maintenance work.

	Valid	Missing	Mean	Median	Mode
% of new development work	42	2	60.4762	62.5000	60.00*
% of maintenance work	41	3	40.4878	40.0000	20.00*

* Multiple modes exist. The smallest value is shown

Table 7-19 Percentage of New & Maintenance Work

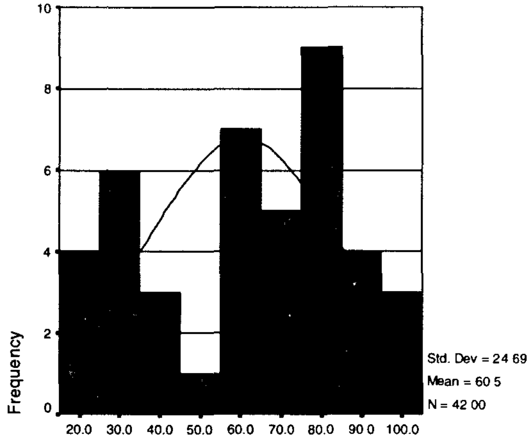

Figure 7-22 Percentage of New Development Work

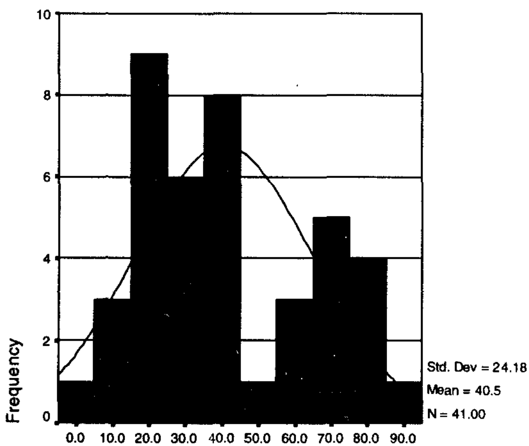

Figure 7-23 Percentage of Maintenance Work

The importance of estimating in the organisation was determined and the results are shown in Table 7-20 and Figure 7-24. The majority, 81.4% advised estimating was important to extremely important to the organisation. A minority, 18.6% advised it was less than important and two people advised it was not important.

	Rating	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	2	4.5	4.7	4.7
	3	6	13.6	14.0	18.6
	4	5	11.4	11.6	30.2
	5	5	11.4	11.6	41.9
	6	12	27.3	27.9	69.8
	7	13	29.5	30.2	100.0
Total		43	97.7	100.0	
Missing		1	2.3		
Total		44	100.0		

Table 7-20 Importance of Estimating in the Organisation

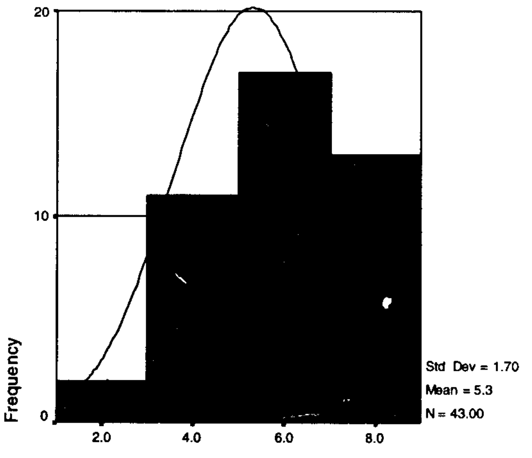

Figure 7-24 Importance of Estimating in the Organisation

The degree of satisfaction expressed with the organisation's estimating process (see Table 7-21) overall was slightly more than satisfied with a mean of 4.3.

	Rating	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	1	2.3	2.3	2.3
	2	2	4.5	4.7	7.0
	3	8	18.2	18.6	25.6
	4	15	34.1	34.9	60.5
	5	10	22.7	23.3	83.7
	6	5	11.4	11.6	95.3
	7	2	4.5	4.7	100.0
Total		43	97.7	100.0	
Missing		1	2.3		
Total		44	100.0		

Table 7-21 Degree of Satisfaction with the Estimating Process

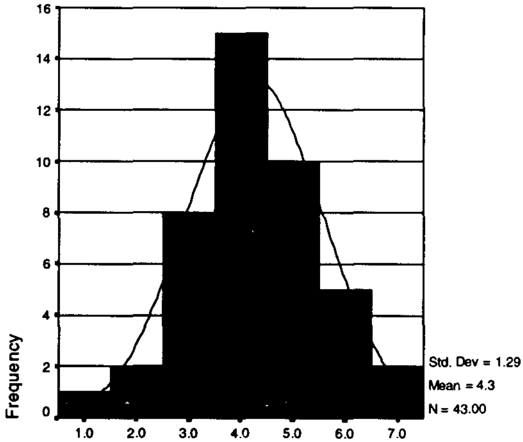

Figure 7-25 Degree of Satisfaction with the Estimating Process

Information relating to the accuracy of the organisations' estimates is detailed in Table 7-22 and Figure 7-26. These show that the estimates are considered slightly more than somewhat accurate with a mean of 4.3.

	Rating	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	3	6.8	7.0	7.0
	3	7	15.9	16.3	23.3
	4	13	29.5	30.2	53.5
	5	12	27.3	27.9	81.4
	6	8	18.2	18.6	100.0
Total		43	97.7	100.0	
Missing		1	2.3		
Total		44	100.0		

Table 7-22 Accuracy of the Organisation's Estimates

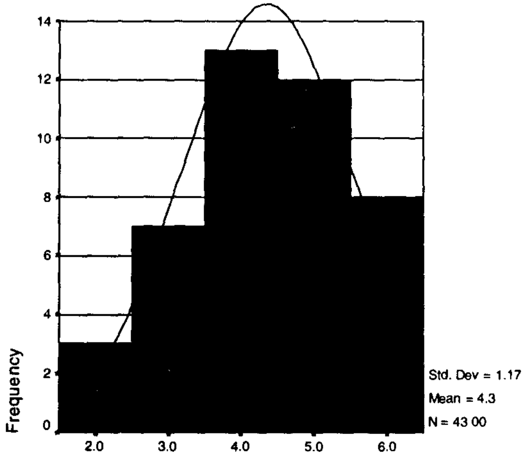

Figure 7-26 Accuracy of the Organisation's Estimates

Data concerning the reasons initial estimates are required was gathered. This information was required, in part, to substantiate the hypothesis that estimates are typically required before the detailed requirements are known. Table 7-23 clearly shows this to be the case, for instance 65% of the respondents advised they undertook estimating to obtain project approval. This adds weight to the proposal that estimating techniques are needed that formalise WBS, analogy etc.

Reasons	Valid	Valid %	Missing
To provide firm quotations	28	63.6	16
Organisational requirement	14	31.8	30
To obtain project approval	29	65.9	15
Assess project risk	22	50.0	22
Budgeting	33	75.0	11
Project planning	7	15.9	37

Table 7-23 Reasons for Estimating

Table 7-24 provides information on the number of estimates made in the respondent's organisation in the last year. It is interesting to note that the estimates required are divided equally between new and maintenance projects. This can be seen with the median and mode being ten in all cases. One response indicated they conducted two thousand maintenance estimates and appears to be contrary to all the other responses.

	Valid	Missing	Mean	Median	Mode	Total
No of new project estimates made in the last year	38	6	19.71	10.00	10	749
No of maintenance project estimates made in the last year	39	5	70.44	10.00	10	2747

Table 7-24 No of Estimates made in Last Year

Chapter 8 Elements Estimated: WA Current Practice

A number of elements, that would contribute to the overall project estimate, were listed and the respondents were asked to advise whether the elements were considered or not. The elements listed consisted of the lifecycle phases and other significant activities together with items that could result in major purchases and would also impact on the amount of project work required. They had the choice of never, sometimes or always. They were also asked to indicate their degree of confidence in the estimate for the particular element with one indicating "not confident", four "somewhat confident" and seven "extremely confident".

8.1 Problem Definition & Feasibility Study

For the element "Problem Definition & Feasibility Study", the Figure 8-1 shows that this phase is only never considered by 2.3% of the respondents which corresponds with a single response. It is sometimes considered by 43.2% and always considered by 52.3%. This could indicate that a number of organisations do not always estimate this first phase as it is problematic if the problem and hence the scope is unknown. The degree of confidence (see Table 8-1 and Figure 8-2) is generally high with a mean of 4.8 although a significant percentage, 21.4% were less than somewhat confident in the estimate made. Of the respondents 64.3% designated five or above on the scale which would indicate they were fairly confident in their estimate for this element.

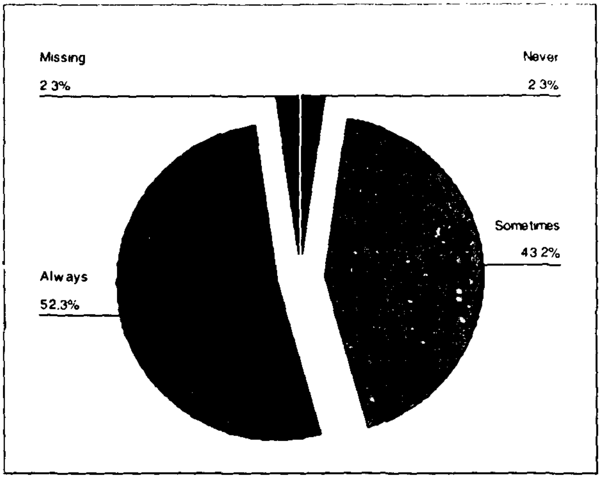

Figure 8-1 Consideration of Problem definition & feasibility study

	Degree of Confidence	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	1	2.3	2.4	2.4
	2	1	2.3	2.4	4.8
	3	7	15.9	16.7	21.4
	4	6	13.6	14.3	35.7
	5	10	22.7	23.8	59.5
	6	15	34.1	35.7	95.2
	7	2	4.5	4.8	100.0
Total		42	95.5	100.0	
System Missing		2	4.5		
Total		44	100.0		

Table 8-1 Degree of confidence - Problem definition & Feasibility Study

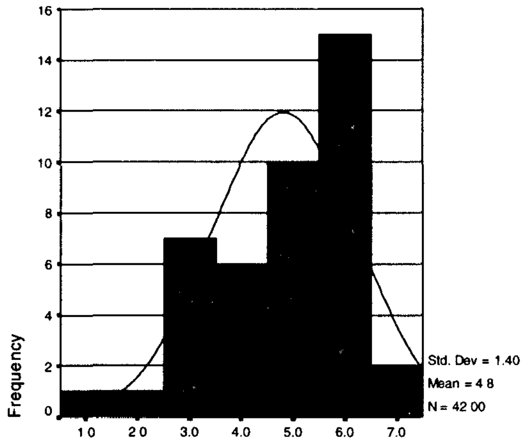

Figure 8-2 Degree of confidence - Problem Definition & feasibility study

8.2 Requirements Analysis

For the element "Requirements Analysis", the Figure 8-3 shows that this phase is "sometimes" considered by 11.4% and always considered by 88.6%. This is the only element that all respondents indicated they estimated. That is there were no missing data and the "never" choice was not designated.

The degree of confidence (see Table 8-2 and Figure 8-4) is very high with a mean of 5.1. A small percentage, 9.1%, were less than somewhat confident in the estimate made. Of the respondents 77.3% designated five or above on the scale which would indicate they were more than somewhat confident in their estimate for this element. This was the second highest element in percentage terms for scores above five. This is the highest element for the number of respondents, thirty four, who indicated they were more than somewhat confident in their estimate.

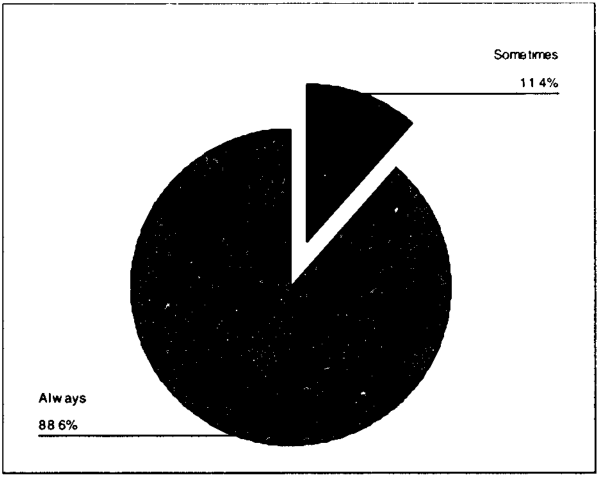

Figure 8-3 Consideration of Requirements analysis

	Degree of Confidence	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	1	2.3	2.3	2.3
	3	3	6.8	6.8	9.1
	4	6	13.6	13.6	22.7
	5	18	40.9	40.9	63.6
	6	14	31.8	31.8	95.5
	7	2	4.5	4.5	100.0
Total		44	100.0	100.0	

Table 8-2 Degree of confidence - Requirements analysis

five. This is the equal second highest element for the number of respondents, thirty one, who indicated they were more than somewhat confident in their estimate.

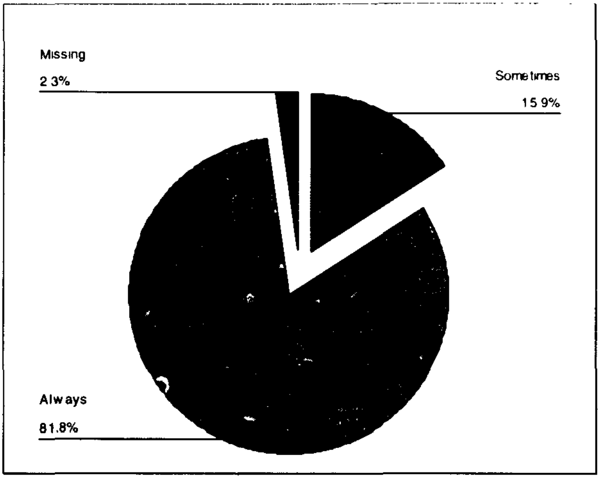

Figure 8-5 Consideration of Design

	Degree of Confidence	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	3	3	6.8	7.0	7.0
	4	9	20.5	20.9	27.9
	5	18	40.9	41.9	69.8
	6	11	25.0	25.6	95.3
	7	2	4.5	4.7	100.0
	Total	43	97.7	100.0	
System Missing		1	2.3		
Total		44	100.0		

Table 8-3 Degree of confidence - Design

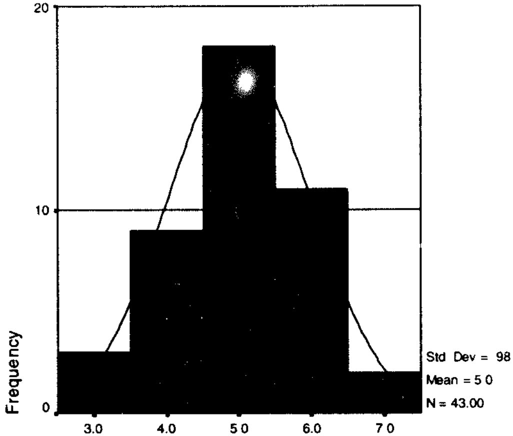

Figure 8-6 Degree of confidence - Design

8.4 Development

For the element "Development", which indicated the coding activities in the survey form, Figure 8-7 shows that this phase is sometimes considered by 11.4%, never considered by 4.5% and one respondent, representing 2.3%, made no indication. It was always considered by 81.8%, which is the same result as for the "Design" element.

The degree of confidence (see Table 8-4 and Figure 8-8) is lower than requirements and design with a mean of 4.7 and 17.1% were less than somewhat confident in the estimate made. It is interesting to note only 56.1% or twenty-three of the respondents designated five or above on the scale which would indicate they were more than somewhat confident in their estimate for this element. This would tend to indicate that people in general are only somewhat confident in their estimates for the coding phase, which is the lowest level for the classic lifecycle phases excluding maintenance.

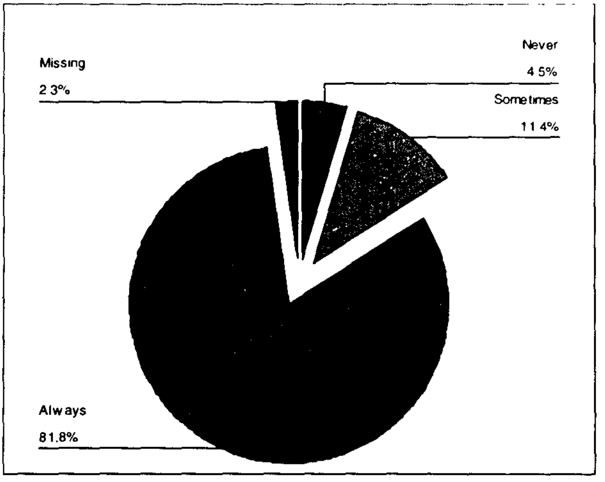

Figure 8-7 Consideration of Developement

	Degree of Confidence	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	2	4.5	4.9	4.9
	3	3	6.8	7.3	12.2
	4	13	29.5	31.7	43.9
	5	12	27.3	29.3	73.2
	6	8	18.2	19.5	92.7
	7	3	6.8	7.3	100.0
Total		41	93.2	100.0	
System Missing		3	6.8		
Total		44	100.0		

Table 8-4 Degree of confidence - Development

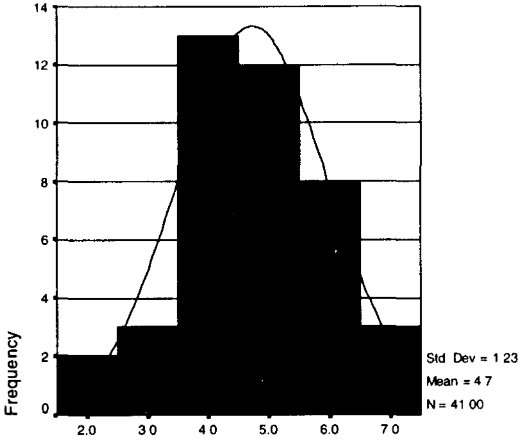

Figure 8-8 Degree of confidence - Development

8.5 Implementation

For the element "Implementation", Figure 8-9 shows that this phase is sometimes considered by 11.4%, never considered by 6.8% and one respondent, representing 2.3%, made no indication. It was always considered by 79.8%.

The degree of confidence (see Table 8-5 and Figure 8-10) is lower than requirements and design with a mean of 4.8 and marginally higher than the confidence in the "Development" element. Of the respondents 17.1% were less than somewhat confident in the estimate made which is the same as the "Development" element. Twenty-five or 61% of the respondents designated five or above on the scale, which would indicate, they were more than somewhat confident in their estimate for this element. This would tend to indicate that people in general are only somewhat confident in their estimates for the implementation phase. The results for this element are comparable with those of the "Development" element.

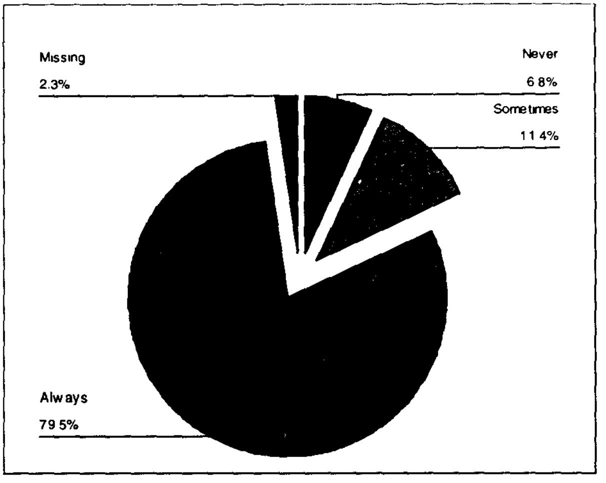

Figure 8-9 Consideration of Implementation

	Degree of Confidence	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	3	6.8	7.3	7.3
	3	1	2.3	2.4	9.8
	4	12	27.3	29.3	39.0
	5	13	29.5	31.7	70.7
	6	8	18.2	19.5	90.2
	7	4	9.1	9.8	100.0
Total		41	93.2	100.0	
System Missing		3	6.8		
Total		44	100.0		

Table 8-5 Degree of confidence - Implementation

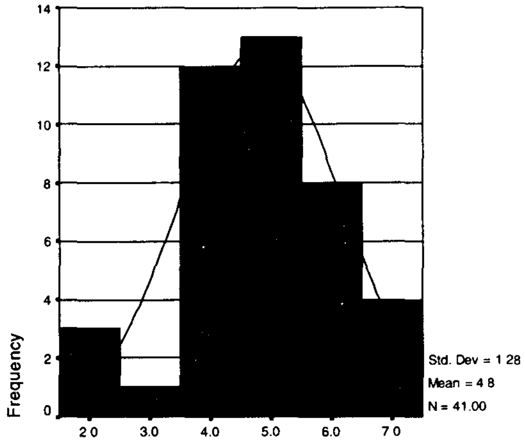

Figure 8-10 Degree of confidence - Implementation

8.6 Training

For the element "Training", Figure 8-11 shows that this phase is sometimes considered by 52.3%, never considered by 4.5% and two respondents, representing 4.5%, made no indication. It was always considered by 38.6%.

The degree of confidence (see Table 8-6 and Figure 8-12) with a mean of 4.8 indicates the people who made estimates for this element were somewhat confident with their estimate. Of the respondents, 14.6% were less than somewhat confident in the estimate made. Twenty-five or 60.9% of the respondents designated five or above on the scale, which would indicate they were more than somewhat confident in their estimate for this element. This would tend to indicate that people in general are only somewhat confident in their estimates for "Training".

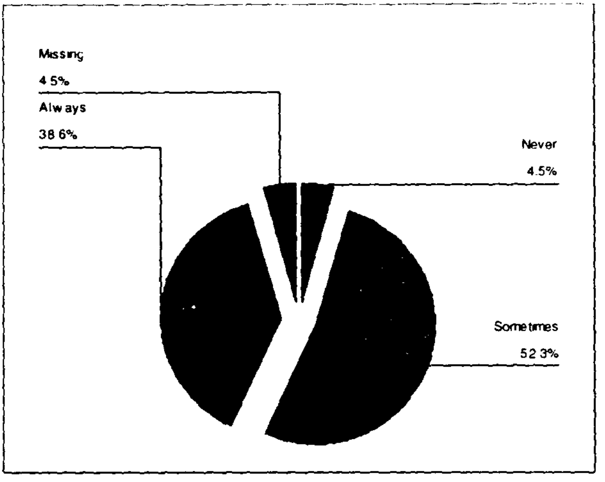

Figure 8-11 Consideration of Training

	Degree of Confidence	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	1	2.3	2.4	2.4
	3	4	9.1	9.8	12.2
	4	11	25.0	26.8	39.0
	5	11	25.0	26.8	65.9
	6	13	29.5	31.7	97.6
	7	1	2.3	2.4	100.0
Total		41	93.2	100.0	
System Missing		3	6.8		
Total		44	100.0		

Table 8-6 Degree of confidence - Training

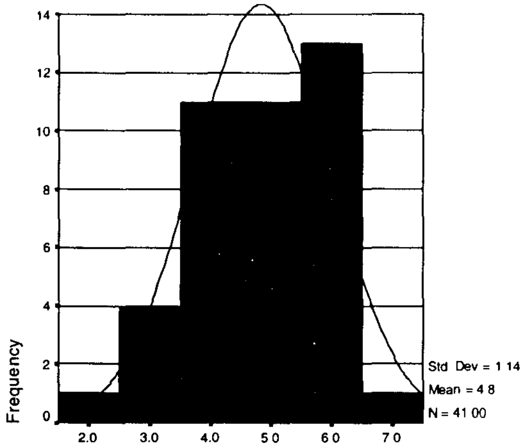

Figure 8-12 Degree of confidence - Training

8.7 Project Management & Administration

For the element "Project Management & Administration", Figure 8-13 shows that this phase is sometimes considered by 25%, never considered by 9.1% and two respondents, representing 4.5%, made no indication. It was always considered by 61.4%.

The degree of confidence (see Table 8-7 and Figure 8-14) with a mean of 5.1 is quite high. Two respondents or 5.1% were less than somewhat confident in the estimate made. Thirty-one or 79.4% of the respondents designated five or above on the scale which would indicate they were more than somewhat confident in their estimate for this element. The results would tend to indicate that people in general are quite confident in their estimates, when made, for "Project Management & Implementation".

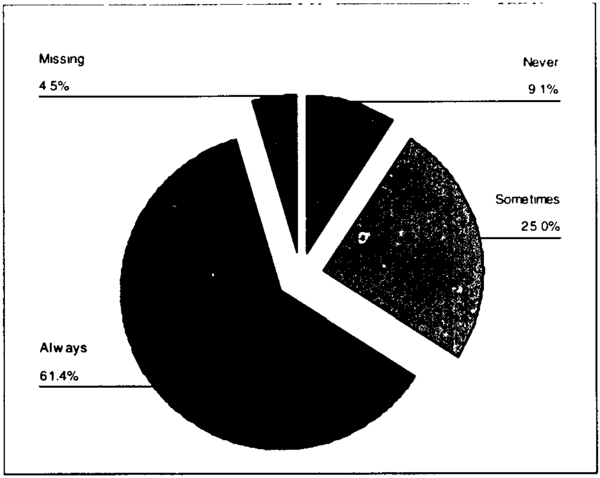

Figure 8-13 Consideration of Project management & administration

	Degree of Confidence	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	1	2.3	2.6	2.6
	3	1	2.3	2.6	5.1
	4	6	13.6	15.4	20.5
	5	16	36.4	41.0	61.5
	6	13	29.5	33.3	94.9
	7	2	4.5	5.1	100.0
Total		39	88.6	100.0	
System Missing		5	11.4		
Total		44	100.0		

Table 8-7 Degree of confidence - Project management & administration

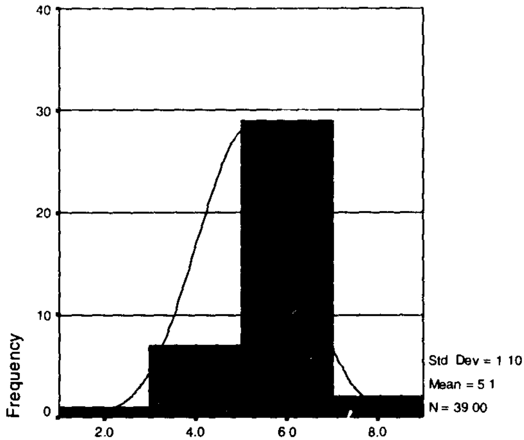

Figure 8-14 Degree of confidence - Project management & administration

8.8 Development Hardware & Software

For the element "Development Hardware & Software", Figure 8-15 shows that this phase is never considered by 11.4% of the respondents. It is sometimes considered by 47.7% and three respondents, representing 6.8%, made no indication. It was always considered by 34.1% of the respondents.

The degree of confidence (see Table 8-8 and Figure 8-16) is generally high with a mean of five although 13.9% were less than somewhat confident in the estimate made. Twenty-three or 63.9% of the respondents designated five or above on the scale which would indicate they were confident in their estimate for this element.

Six of the respondents representing 16.7% indicated they were extremely confident in the estimates made which is significantly greater than most of the other elements.

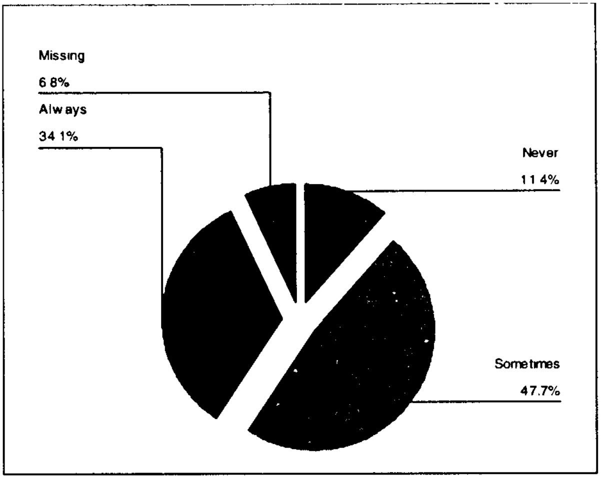

Figure 8-15 Consideration of Development hardware & software

	Degree of Confidence	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	1	2.3	2.8	2.8
	3	4	9.1	11.1	13.9
	4	8	18.2	22.2	36.1
	5	10	22.7	27.8	63.9
	6	7	15.9	19.4	83.3
	7	6	13.6	16.7	100.0
Total		36	81.8	100.0	
System Missing		8	18.2		
Total		44	100.0		

Table 8-8 Degree of confidence - Development hardware & software

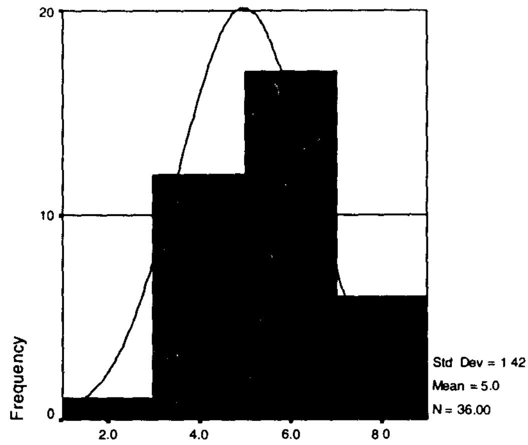

Figure 8-16 Degree of confidence - Development hardware & software

8.9 Operational Hardware & Software

For the element "Operational Hardware & Software", the Figure 8-17 shows that this phase is never considered by 13.6% of the respondents. It is sometimes considered by 40.9% and three respondents, representing 6.8%, made no indication. It was always considered by 38.6% of the respondents.

The degree of confidence (see Table 8-9 and Figure 8-18) is generally high with a mean of five although 11.8% were less than somewhat confident in the estimate made. Twenty-two or 64.7% of the respondents designated five or above on the scale which would indicate they were confident in their estimate for this element. Five of the respondents representing 14.7% percent indicated they were extremely confident in the estimates made which is significantly greater than most of the other elements.

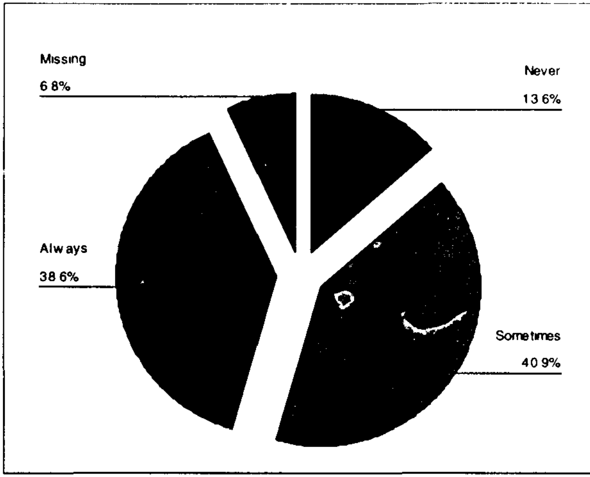

Figure 8-17 Consideration of Operational hardware & software

	Degree of Confidence	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	1	2.3	2.9	2.9
	3	3	6.8	8.8	11.8
	4	8	18.2	23.5	35.3
	5	10	22.7	29.4	64.7
	6	7	15.9	20.6	85.3
	7	5	11.4	14.7	100.0
Total		34	77.3	100.0	
System Missing		10	22.7		
Total		44	100.0		

Table 8-9 Degree of confidence - Operational Hardware & software

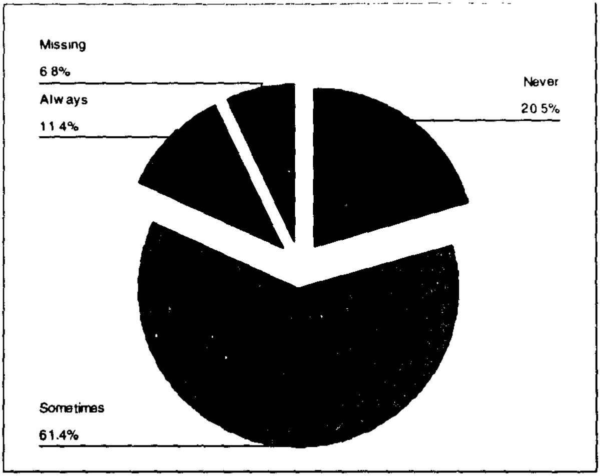

Figure 8-19 Consideration of Environmental changes

	Degree of Confidence	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	2	4.5	5.7	5.7
	2	5	11.4	14.3	20.0
	3	6	13.6	17.1	37.1
	4	11	25.0	31.4	68.6
	5	9	20.5	25.7	94.3
	6	1	2.3	2.9	97.1
	7	1	2.3	2.9	100.0
Total		35	79.5	100.0	
System Missing		9	20.5		
Total		44	100.0		

Table 8-10 Degree of confidence - Environmental changes

Eleven of the respondents or 31.5% designated five or above on the scale, which would indicate they were confident in their estimate for this element. This was the least element considered, with the exception of "Other", and for the people who made an estimate they were generally less that somewhat confident in their estimate.

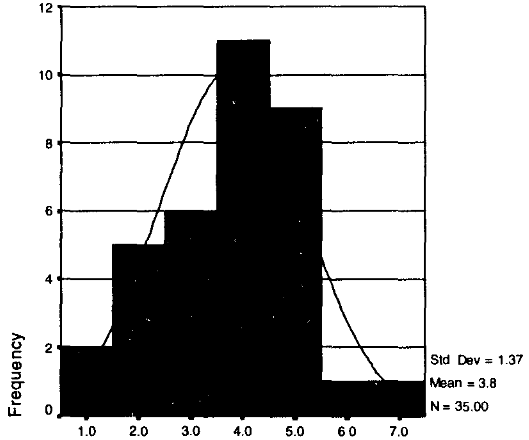

Figure 8-20 Degree of confidence - Environmental changes

8.11 Maintenance

For the element "Maintenance", Figure 8-21 shows that this phase is never considered by 15.9% of the respondents. It is sometimes considered by 52.3% and four respondents, representing 9.1%, made no indication. It was always considered by 22.7% of the respondents which is one of the lowest of the elements always considered.

The degree of confidence (see Table 8-11 and Figure 8-22) with a mean of 4.3 is towards the low end of the scale in comparison to all the other elements estimated.

Twenty percent were less than somewhat confident in the estimate made. Nineteen or 54.3% of the respondents designated five or above on the scale which would indicate they were confident in their estimate for this element.

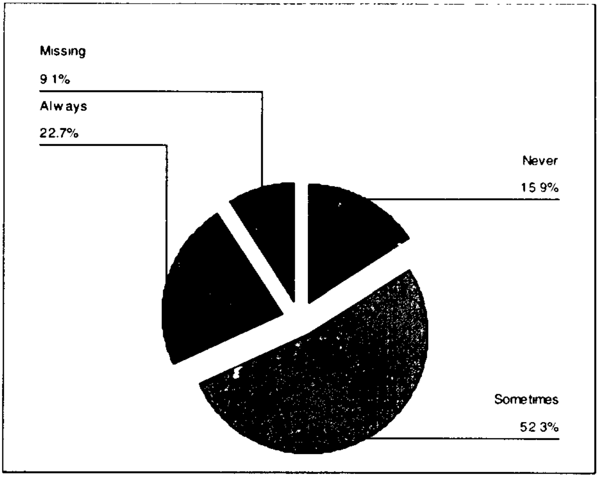

Figure 8-21 Consideration of Maintenance

Eight of the respondents representing 18.2% indicated they were extremely confident in the estimates made which was the highest of all elements considered.

It should be noted this element also had the largest standard deviation at 1.47. This would indicate that whilst some of the respondents were very confident in their estimates others were not.

	Degree of Confidence	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	2	4.5	5.7	5.7
	2	4	9.1	11.4	17.1
	3	1	2.3	2.9	20.0
	4	9	20.5	25.7	45.7
	5	11	25.0	31.4	77.1
	6	8	18.2	22.9	100.0
Total		35	79.5	100.0	
System Missing		9	20.5		
Total		44	100.0		

Table 8-11 Degree of confidence - Maintenance

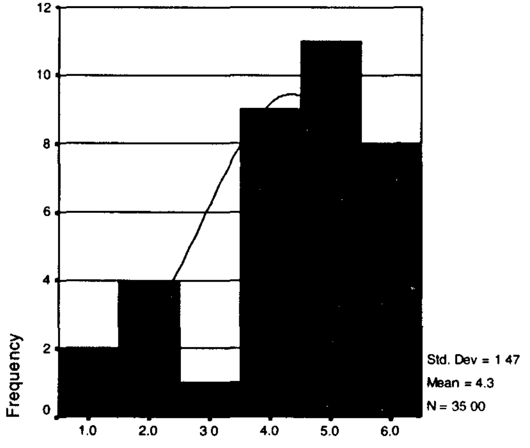

Figure 8-22 Degree of confidence - Maintenance

8.12 Other

An "Other" category was available for the respondents to indicate elements that were not listed. The Figure 8-23 shows that for "other" elements 77.3% of the respondents were in the missing category. As there was no "never" choice this figure shows the elements listed in the questionnaire covered the estimating scope of most respondents. Other elements are sometimes considered by 15.9% and was always considered by 6.8% of the respondents which is the lowest of the elements always considered.

The degree of confidence (see Table 8-12 and Figure 8-24) with a mean of 4.3 is towards the low end of the scale in comparison to all the other elements estimated. Thirty percent were less than somewhat confident in the estimate made. Fifty percent representing five of the respondents designated five or above on the scale which would indicate they were confident in their estimate for the element they designated.

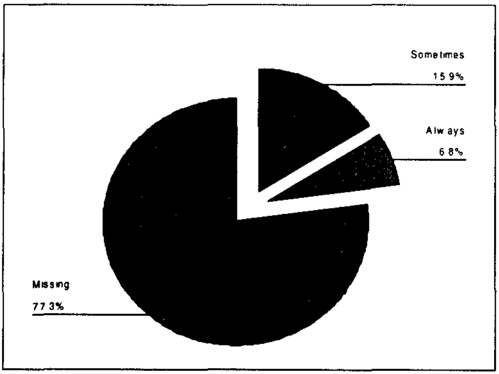

Figure 8-23 Consideration of other

	Degree of Confidence	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	3	3	6.8	30.0	30.0
	4	2	4.5	20.0	50.0
	5	4	9.1	40.0	90.0
	6	1	2.3	10.0	100.0
Total		10	22.7	100.0	
System Missing		34	77.3		
Total		44	100.0		

Table 8-12 Degree of confidence - other

It should be noted no respondent indicated they were extremely confident in the estimates made. The graph in Figure 8-24 shows a clustering around the somewhat confident mark.

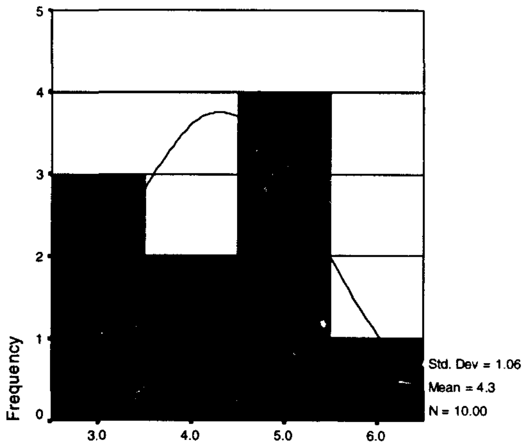

Figure 8-24 Degree of confidence - other

8.13 Conclusions

Table 8-13 gives a summary of some of the significant data in relation to the elements estimated and the degree of confidence the respondents have in those estimates when they are made.

As can be seen the standard lifecycle phases of:

- Requirements Analysis
- Design
- Development (Coding) and
- Implementation

are nearly always considered by approximately 80% or greater of the respondents. The consideration of these elements in forming estimates is significantly higher that all the others proposed in the survey questionnaire. The other element that is considered by a high percentage, sixty-one point four of the respondents is "Project Management & Administration". The other elements are only are nearly always considered by approximately fifty percent or less of the respondents.

An observation that can be made is that the confidence in the estimates diminishes for the later phases in the lifecycle. The confidence in requirements analysis and design estimating is significantly higher than coding, implementation, training and maintenance. Start up activities, such as project management also have a high confidence rating. In general it can be said that where an estimate is made the respondent is more than somewhat confident in the estimate. This could be either an indication of the optimism of the estimators or the degree of expertise and experience of the people who responded to the survey.

It would appear the phases and elements that must be completed and are measurable have a lower confidence. That is phases such as requirements and design can be arbitrarily stopped when sufficient work has been done however their "completeness" cannot be tested. The development, training and equipment purchases however can be tested and costed.

Element	% Who Always Consider	Confidence Mean	% Above Somewhat Confident	Standard Deviation
Problem Defn & Feasibility Study	52.3	4.8	64.3	1.40
Requirements Analysis	88.6	5.1	77.3	1.07
Design	81.8	5.0	72.1	0.98
Development (Coding)	81.8	4.7	56.1	1.23
Implementation	79.5	4.8	61.0	1.28
Training	38.6	4.8	60.9	1.14
Project Management & Administration	61.4	5.1	79.4	1.10
Development Hardware & Software	34.1	5.0	63.9	1.42
Operational Hardware & Software	38.6	5.0	64.7	1.38
Environmental Changes	11.4	3.8	31.5	1.37
Maintenance	22.7	4.3	54.3	1.47
Other	6.8	4.3	50.0	1.06

Table 8-13 Summary of elements estimated & their degree of confidence

The validity of the responses and lack of variance in the measurement of the "Degree of Confidence" is supported with the standard deviation being less than one point five (1.5) for all the data.

As there were few additional elements nominated it would seem the defined elements would be a reasonable starting point for an estimating model. Nominated "Other" elements were considered sub-sets of the defined elements.

Chapter 9 Modifying Parameters: WA Current Practice

A number of parameters were listed which could cause the respondent's nominal estimate to be modified. The listed parameters were derived from cost drivers associated with COCOMO one and two, Function Point Analysis and quality attributes associated with software. The respondents were asked to advise whether the parameters were considered or not and they had the choice of never, sometimes or always. They were also asked to indicate the degree of impact the particular parameter had on the estimate in terms of its significance. They were asked to indicate on a seven point scale with one indicating "not significant", four " significant " and seven "very significant ".

9.1 People skills

For the parameter "People skills", Figure 9-1 shows that this parameter is only never considered by 4.5% of the respondents which corresponds with two responses. It is sometimes considered by 43.2% and always considered by 50%. There is 2.3% missing which equates to a single response.

The degree of impact (see Table 9-1 and Figure 9-2) is very high with a mean of 5.6 although a small percentage, 4.9%, indicated the impact was less than significant. Of the respondents 70.7% designated five or above on the scale which would indicate the degree of impact was more than significant for this parameter. An interesting observation is that whilst only 50% always consider this parameter of those who consider it 13 consider the degree of impact to be very significant. No one considered it not significant. This was the highest number who considered the degree of impact to be very significant, the next highest was eight for "Knowledge of the Problem Domain".

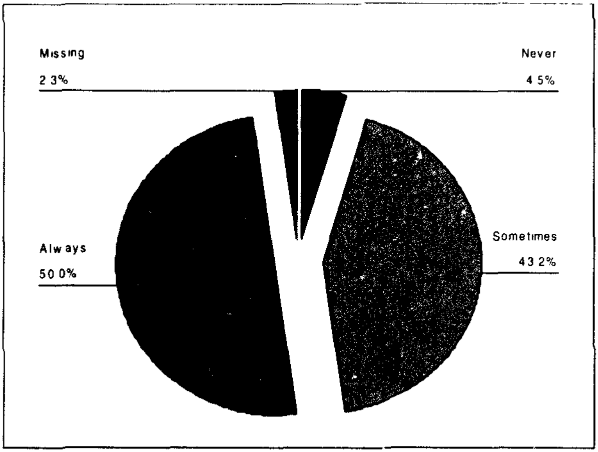

Figure 9-1 Consideration of People skills

	Degree of Impact	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	3	2	4.5	4.9	4.9
	4	10	22.7	24.4	29.3
	5	5	11.4	12.2	41.5
	6	11	25.0	26.8	68.3
	7	13	29.5	31.7	100.0
Total		41	93.2	100.0	
System Missing		3	6.8		
Total		44	100.0		

Table 9-1 Degree of impact - People skills

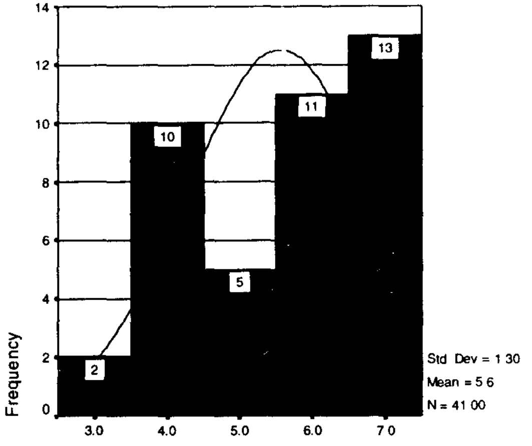

Figure 9-2 Degree of impact - People skills

9.2 Knowledge of the Problem Domain

For the parameter " Knowledge of the Problem Domain ", Figure 9-3 shows that this parameter is only never considered by 4.5% of the respondents which corresponds with two responses. It is sometimes considered by 29.5% and always considered by 61.4%. There is 4.5% missing which equates to two responses.

The degree of impact (see Table 9-2 and Figure 9-4) is very high with a mean of 5.3 although a small percentage, 12.5%, indicated the impact was less than significant. Of the respondents, 72.5% designated five or above on the scale, which would indicate, the degree of impact was more than significant for this parameter. For those who considered "People Skills", eight consider the degree of impact to be very significant and no one considers it not significant. This was the second highest number who considered the degree of impact to be very significant, the highest was thirteen for "People Skills".

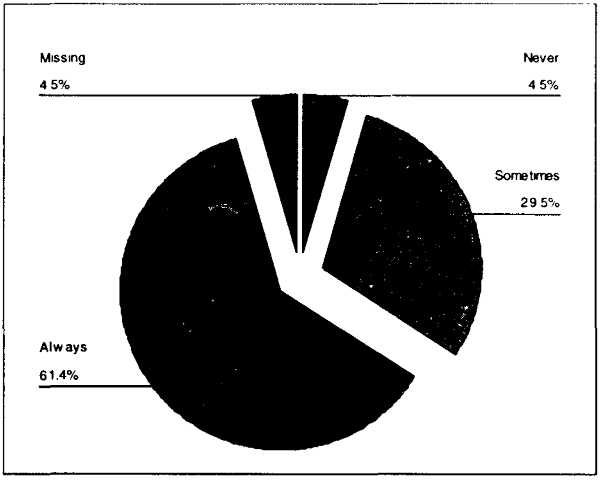

Figure 9-3 Consideration of Knowledge of the application domain

	Degree of Impact	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	1	2.3	2.5	2.5
	3	4	9.1	10.0	12.5
	4	6	13.6	15.0	27.5
	5	10	22.7	25.0	52.5
	6	11	25.0	27.5	80.0
	7	8	18.2	20.0	100.0
Total		40	90.9	100.0	
System Missing		4	9.1		
Total		44	100.0		

Table 9-2 Degree of impact - Knowledge of the application domain

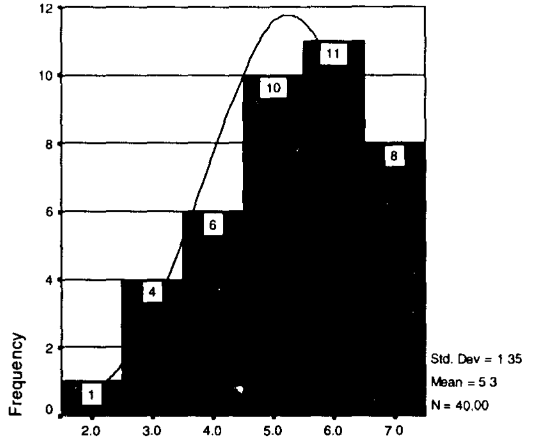

Figure 9-4 Degree of impact - knowledge of the application domain

9.3 Complexity of the Problem

For the parameter " Complexity of the Problem", Figure 9-5 shows that this parameter is only never considered by 4.5% of the respondents which corresponds with two responses. It is sometimes considered by 27.3% and always considered by sixty-three point six percent. Two responses or 4.5% are missing.

The degree of impact (see Table 9-3 and Figure 9-6) is very high with a mean of 5.2 although a small percentage, 7.7%, indicated the impact was less than significant. Of the respondents 79.4% designated five or above on the scale which would indicate the degree of impact was more than significant for this parameter. This was the highest percentage, of all the parameters with the exception of "Others", above significant and hence respondents are indicating this has the greatest impact on their estimates.

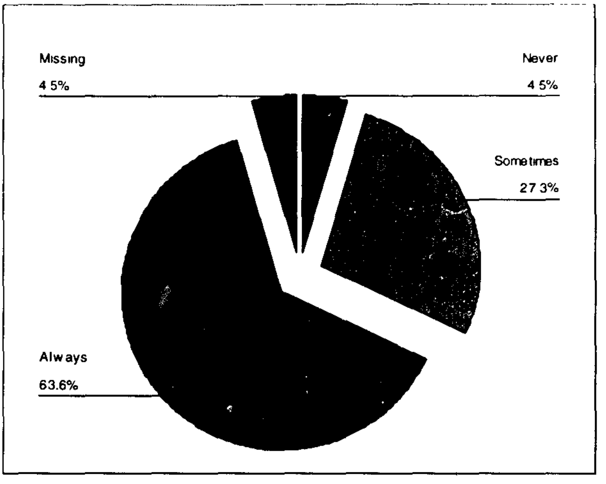

Figure 9-5 Consideration of Complexity of the problem

	Degree of Impact	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	1	2.3	2.6	2.6
	2	1	2.3	2.6	5.1
	3	1	2.3	2.6	7.7
	4	5	11.4	12.8	20.5
	5	13	29.5	33.3	53.8
	6	13	29.5	33.3	87.2
	7	5	11.4	12.8	100.0
Total		39	88.6	100.0	
System Missing		5	11.4		
Total		44	100.0		

Table 9-3 Degree of impact - Complexity of the problem

For those who considered "Complexity of the Problem", five consider the degree of impact to be very significant and one response considers it not significant.

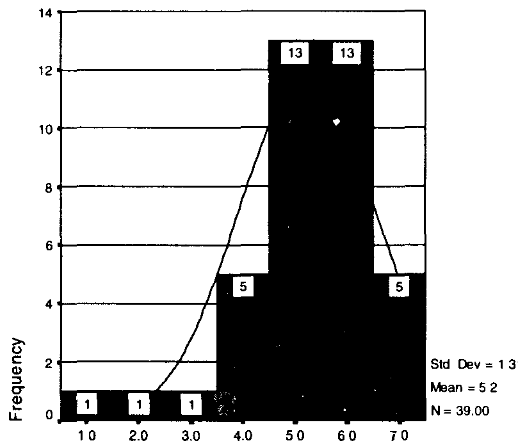

Figure 9-6 Degree of impact - Complexity of the problem

9.4 Algorithmic Complexity

For the parameter " Algorithmic Complexity ", Figure 9-7 shows that this parameter is never considered by 20.5% of the respondents. It is sometimes considered by 56.8% and always considered by 15.9%. This is the lowest "always considered" score, with the exception of "Other" of all the parameters. Three responses or 6.8% are missing.

The degree of impact (see Table 9-4 and Figure 9-8) is low, in comparison to the other parameters, with a mean of four. 24.2%, indicated the impact was less than significant. There was 30.3% of the respondents who designated five or above on the scale which was the second lowest recorded. This would indicate the degree of impact was not as highly significant as the other parameters. It must be noted that a high number, fifteen, indicated the score of four which represented significant. For those who considered "Algorithmic Complexity", no one considered the degree of impact to be very significant and one response considers it not significant.

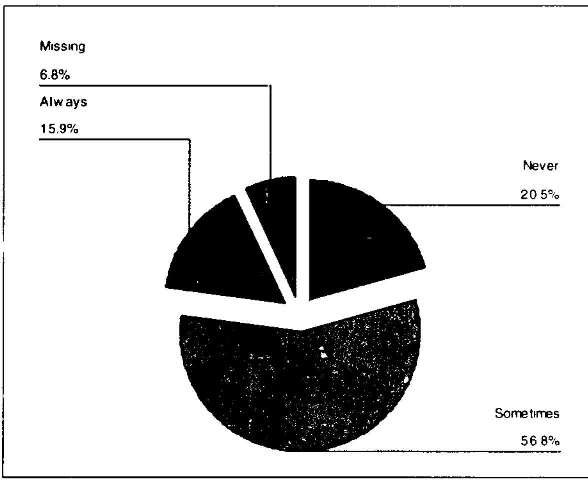

Figure 9-7 Consideration of Algorithmic complexity

	Degree of Impact	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	1	2.3	3.0	3.0
	2	3	6.8	9.1	12.1
	3	4	9.1	12.1	24.2
	4	15	34.1	45.5	69.7
	5	6	13.6	18.2	87.9
	6	4	9.1	12.1	100.0
Total		33	75.0	100.0	
System Missing		11	25.0		
Total		44	100.0		

Table 9-4 Degree of impact - Algorithmic Complexity

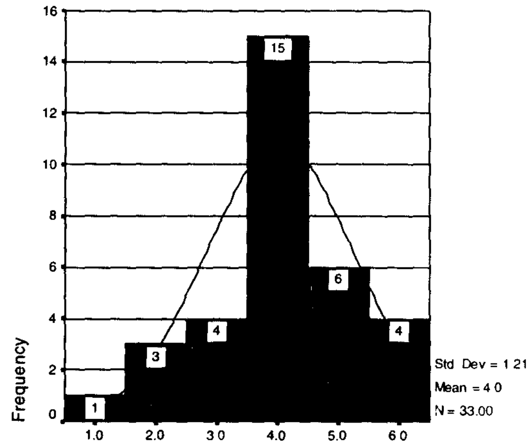

Figure 9-8 Degree of impact - Algorithmic Complexity

9.5 Stability of the Target Platform

For the parameter "Stability of the Target Platform ", Figure 9-9 shows that this parameter is never considered by 15.9% of the respondents. It is sometimes considered by 59.1% and always considered by 20.5%. There are 4.5% missing which equates to two responses.

The degree of impact (see Table 9-5 and Figure 9-10) is comparatively low with a mean of 4.4 and a relatively large percentage, 27.8%, indicated the impact was less than significant. There were 44.4% of the respondents who designated five or above on the scale, which would indicate, the degree of impact was significant for this parameter. For those who considered "Stability of the Target Platform", five consider the degree of impact to be very significant and two responses consider it not significant.

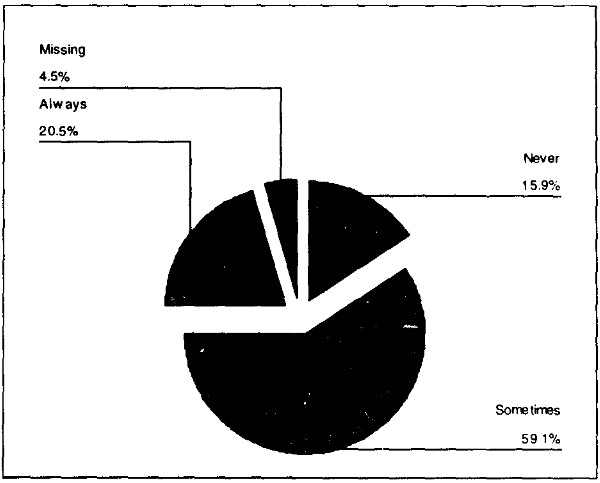

Figure 9-9 Consideration of Stability of the target platform

	Degree of Impact	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	2	4.5	5.6	5.6
	2	2	4.5	5.6	11.1
	3	6	13.6	16.7	27.8
	4	10	22.7	27.8	55.6
	5	7	15.9	19.4	75.0
	6	4	9.1	11.1	86.1
	7	5	11.4	13.9	100.0
Total		36	81.8	100.0	
System Missing		8	18.2		
Total		44	100.0		

Table 9-5 Degree of impact - Stability of the target platform

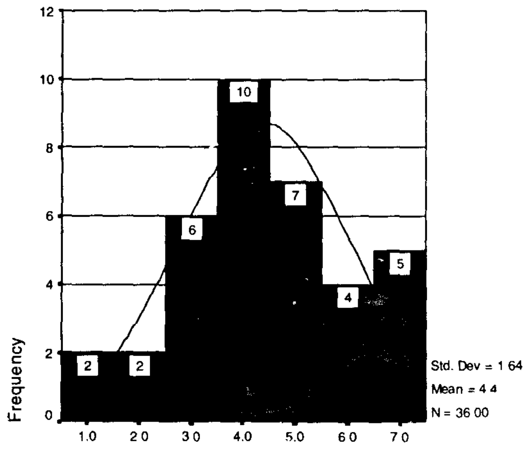

Figure 9-10 Degree of impact - Stability of the target platform

9.6 User Support

For the parameter "User Support", Figure 9-11 shows that this parameter is never considered by 11.4% of the respondents. It is sometimes considered by 50% and always considered by 31.8%. There is 6.8% missing which equates to three responses.

The degree of impact (see Table 9-6 and Figure 9-12) is significant with a mean of 4.8 and 13.5% indicated the impact was less than significant. There were 56.1% of the respondents who designated five or above on the scale which would indicate the degree of impact was significant for this parameter. For those who considered "User Support", three consider the degree of impact to be very significant and no one considered it not significant.

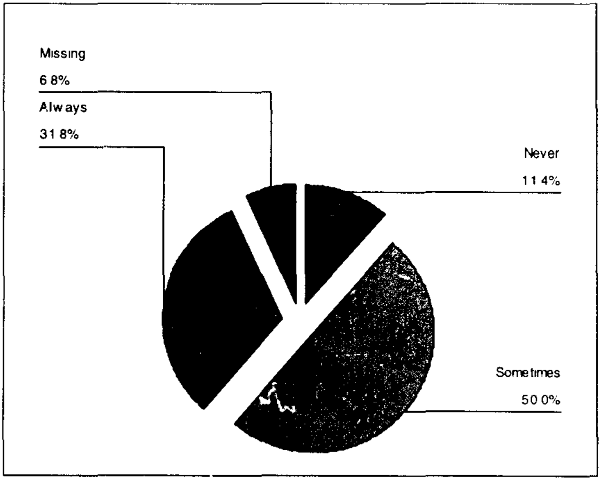

Figure 9-11 Consideration of User support

	Degree of Impact	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	1	2.3	2.7	2.7
	3	4	9.1	10.8	13.5
	4	11	25.0	29.7	43.2
	5	11	25.0	29.7	73.0
	6	7	15.9	18.9	91.9
	7	3	6.8	8.1	100.0
Total		37	84.1	100.0	
System Missing		7	15.9		
Total		44	100.0		

Table 9-6 Degree of Impact - User support

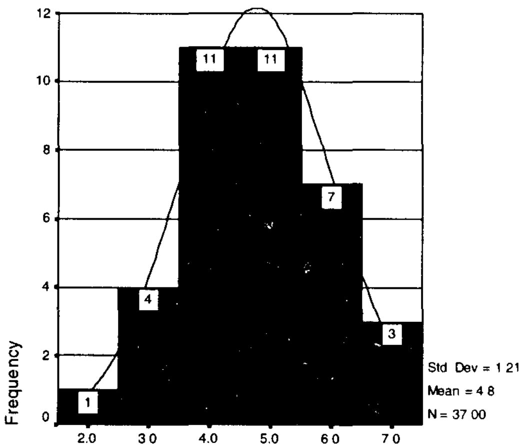

Figure 9-12 Degree of impact - User support

9.7 Training

For the parameter "Training", Figure 9-13 shows that this parameter is never considered by 9.1% of the respondents. It is sometimes considered by 54.5% and always considered by 31.8%. There is 4.5% missing which equates to two responses.

The degree of impact (see Table 9-7 and Figure 9-14) is relatively low with a mean of 4.3 and 23.7% indicated the impact was less than significant. There were 42 2% of the respondents who designated five or above on the scale which would indicate the degree of impact was significant for this parameter. For those who considered "Training", two consider the degree of impact to be very significant and one considered it not significant.

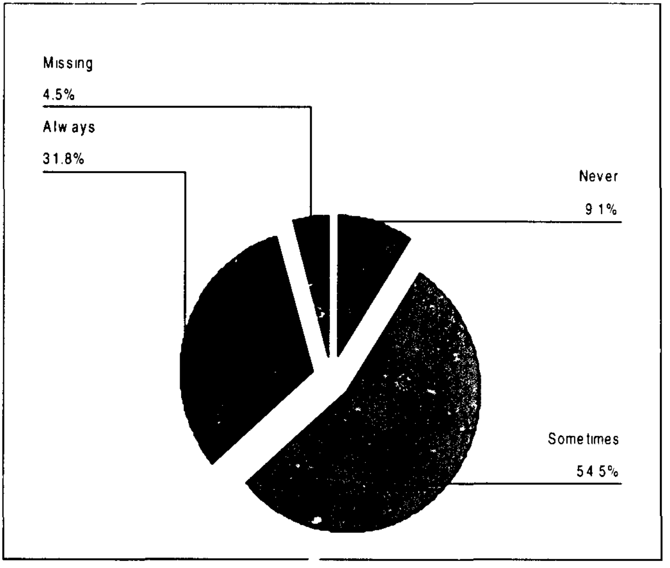

Figure 9-13 Consideration of Training

	Degree of Impact	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	1	2.3	2.6	2.6
	2	2	4.5	5.3	7.9
	3	6	13.6	15.8	23.7
	4	13	29.5	34.2	57 9
	5	8	18 2	21.1	78.9
	6	6	13.6	15.8	94.7
	7	2	4.5	5.3	100.0
Total		38	86.4	100.0	
System Missing		6	13.6		
Total		44	100.0		

Table 9-7 Degree of impact - Training

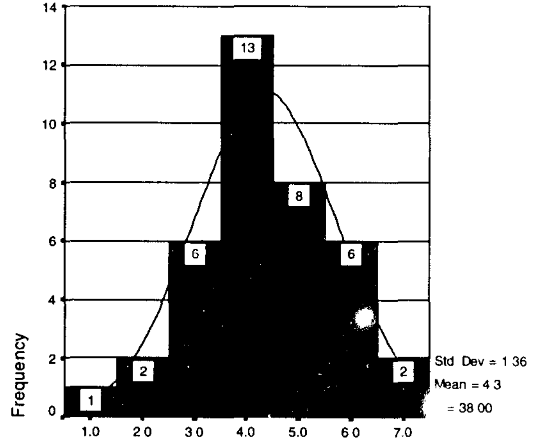

Figure 9-14 Degree of impact - Training

9.8 Maintainability

For the parameter "Maintainability", Figure 9-15 shows that this parameter is never considered by 6.8% of the respondents. It is sometimes considered by 54.5% and always considered by 31.1%. There are 4.5% percent missing which equates to two responses.

The degree of impact (see Table 9-8 and Figure 9-16) is relatively low with a mean of 4.1 and 42% indicated the impact was less than significant. There were 39.4% of the respondents who designated five or above on the scale which would indicate the degree of impact was significant for this parameter. For those who considered "Maintainability", one considered the degree of impact to be very significant and no one considered it not significant.

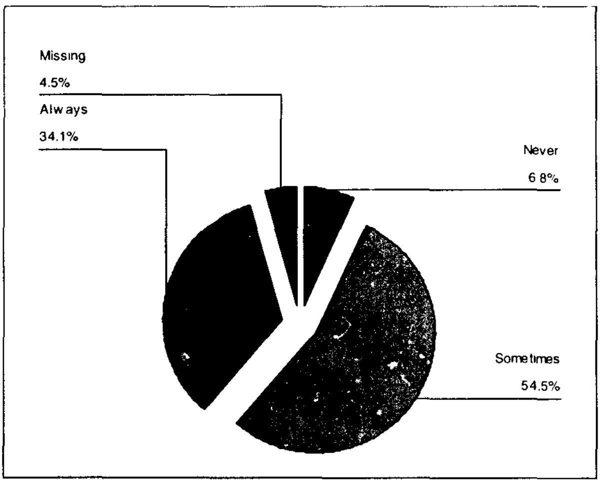

Figure 9-15 Consideration of Maintainability

	Degree of Impact	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	4	9.1	10.5	10.5
	3	12	27.3	31.6	42.1
	4	7	15.9	18.4	60.5
	5	7	15.9	18.4	78.9
	6	7	15.9	18.4	97.4
	7	1	2.3	2.6	100.0
Total		38	86.4	100.0	
System Missing		6	13.6		
Total		44	100.0		

Table 9-8 Degree of impact - Maintainability

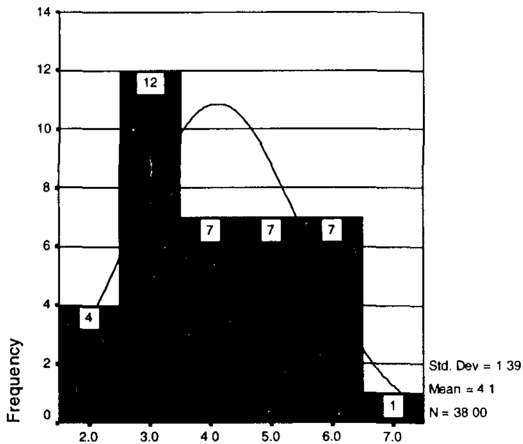

Figure 9-16 Degree of impact - Maintainability

9.9 Performance Requirements

For the parameter "Performance Requirements", Figure 9-17 shows that this parameter is never considered by 4.5% of the respondents. It is sometimes considered by 56.8% and always considered by 36.4%. There is 2.3% missing which equates to one response.

The degree of impact (see Table 9-9 and Figure 9-18) is significant with a mean of 4.7 and 12.5% indicated the impact was less than significant. There were 45 5% of the respondents designated five or above on the scale which would indicate the degree of impact was significant for this parameter. For those who considered "Performance Requirements", two considered the degree of impact to be very significant and no one considered it not significant.

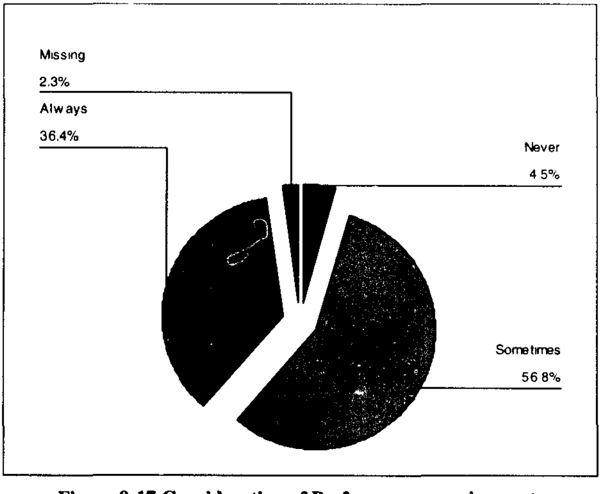

Figure 9-17 Consideration of Performance requirements

	Degree of Impact	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	3	5	11.4	12.5	12.5
	4	17	38.6	42.5	55.0
	5	7	15.9	17.5	72.5
	6	9	20.5	22.5	95.0
	7	2	4.5	5.0	100.0
Total		40	90.9	100.0	
System Missing		4	9.1		
Total		44	100.0		

Table 9-9 Degree of impact - Performance Requirements

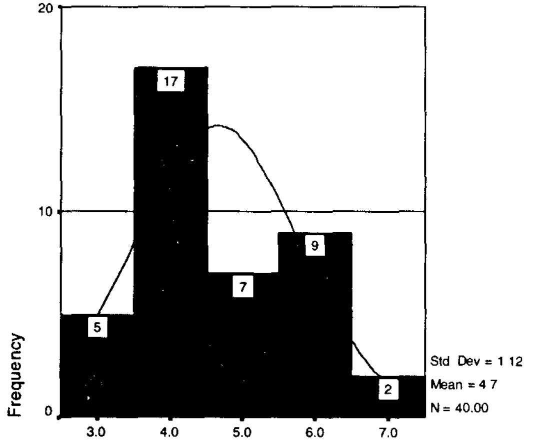

Figure 9-18 Degree of impact - Performance requirements

9.10 Useability

For the parameter "Useability", Figure 9-19 shows that this parameter is never considered by 9.1% of the respondents. It is sometimes considered by 43.2% and always considered by 40.9%. There is 6.8% missing which equates to three responses.

The degree of impact (see Table 9-10 and Figure 9-20) is significant with a mean of 4.6 and 16.2% indicated the impact was less than significant. There were 48.6% of the respondents who designated five or above on the scale that would indicate the degree of impact was significant for this parameter. For those who considered "Useability", three considered the degree of impact to be very significant and one considered it not significant.

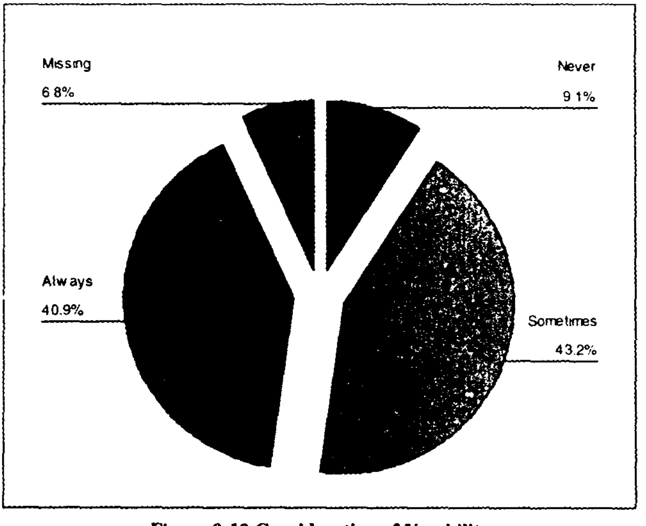

Figure 9-19 Consideration of Useability

	Degree of Impact	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	1	2.3	2.7	2.7
	2	2	4.5	5.4	8.1
	3	3	6.8	8.1	16.2
	4	13	29.5	35.1	51.4
	5	7	15.9	18.9	70.3
	6	8	18.2	21.6	91.9
	7	3	6.8	8.1	100.0
Total		37	84.1	100.0	
System Missing		7	15.9		
Total		44	100.0		

Table 9-10 Degree of impact - Usability

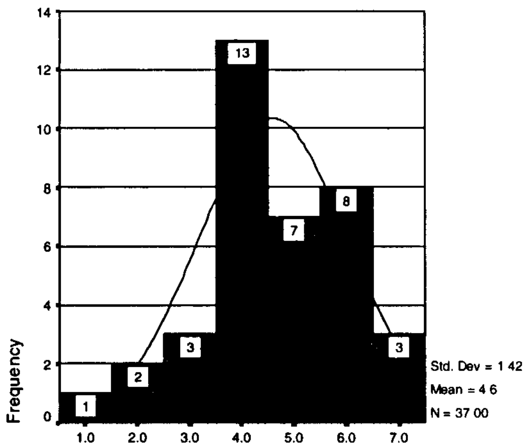

Figure 9-20 Degree of impact - Usability

9.11 Database Size

For the parameter "Database Size", Figure 9-21 shows that this parameter is never considered by 11.4% of the respondents. It is sometimes considered by 47.7% and always considered by 36.4% percent. There is 4.5% missing which equates to two responses.

The degree of impact (see Table 9-11 and Figure 9-22) has a relatively low significance with a mean of 4 and 42.1% indicated the impact was less than significant. There were 31.6% of the respondents who designated five or above on the scale that would indicate the degree of impact was not as significant for this parameter as others. This was the second lowest score. For those who considered "Database size", three considered the degree of impact to be very significant and one considered it not significant.

In terms of the technique, Function Point Analysis, the results above are significant because the highest rated element in the technique relates to the database size. That is the number of logical files and their elements receive the highest weighting when forming a function point count. Refer to Table 2-1 for details of the function point elements and their respective weightings.

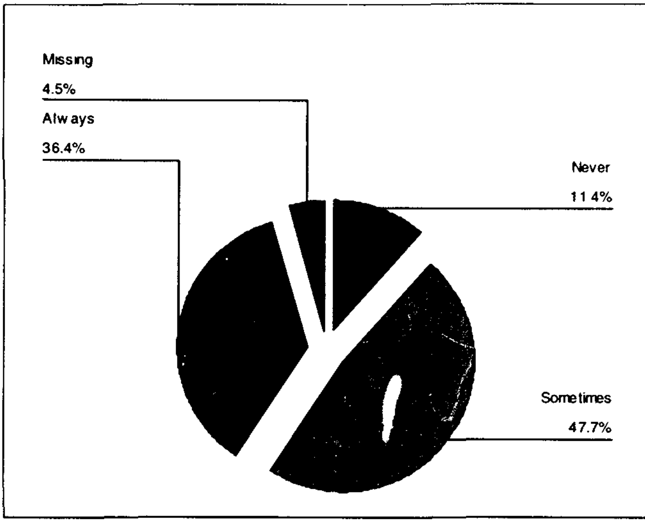

Figure 9-21 Consideration of Database size

	Degree of Impact	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	1	2.3	2.6	2.6
	2	5	11.4	13.2	15.8
	3	10	22.7	26.3	42.1
	4	10	22.7	26.3	68.4
	5	4	9.1	10.5	78.9
	6	5	11.4	13.2	92.1
	7	3	6.8	7.9	100.0
Total		38	86.4	100.0	
System Missing		6	13.6		
Total		44	100.0		

Table 9-11 Degree of impact - Database size

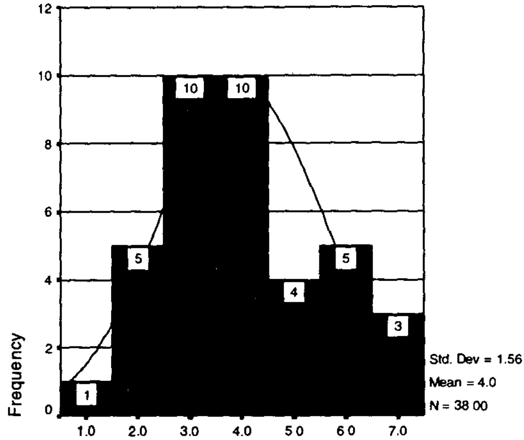

Figure 9-22 Degree of impact - Database size

9.12 Language to be Used

For the parameter "Language to be Used", Figure 9-23 shows that this parameter is never considered by 11.4% of the respondents. It is sometimes considered by 38.6% and always considered by 43.2%. There is 6.8% missing which equates to three responses.

The degree of impact (see Table 9-12 and Figure 9-24) has a relatively low significance with a mean of 4.3 and 27.8% percent, indicated the impact was less than significant. There were 41.6% of the respondents designated five or above on the scale that would indicate the degree of impact was not as significant for this parameter as others. For those who considered "Language to be Used", three considered the degree of impact to be very significant and two considered it not significant.

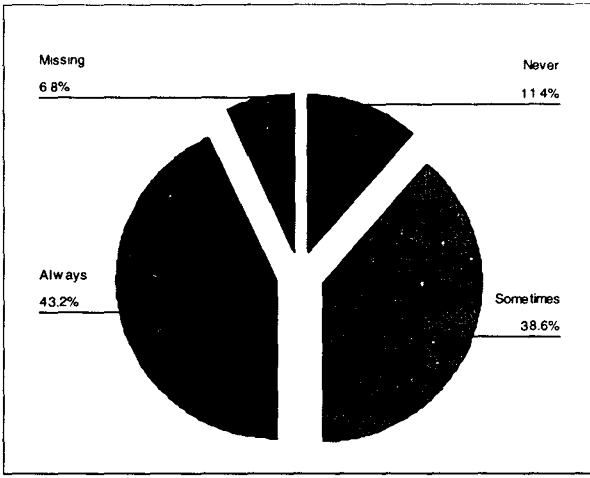

Figure 9-23 Consideration of Language to be used

	Degree of Impact	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	2	4.5	5.6	5.6
	2	2	4.5	5.6	11.1
	3	6	13.6	16.7	27.8
	4	11	25.0	30.6	58.3
	5	7	15.9	19.4	77.8
	6	5	11.4	13.9	91.7
	7	3	6.8	8.3	100.0
Total		36	81.8	100.0	
System Missing		8	18.2		
Total		44	100.0		

Table 9-12 Degree of impact - Language to be used

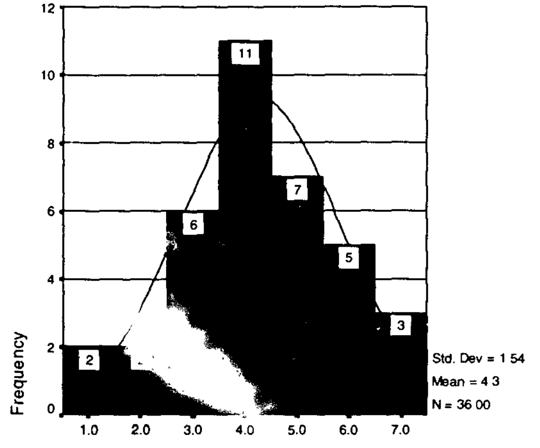

Figure 9-24 Degree of impact - Language to be used

9.13 Reliability

For the parameter "Reliability", Figure 9-25 shows that this parameter is never considered by 11.4% of the respondents. It is sometimes considered by 38.6% and always considered by 43.2%. There is 6.8% missing which equates to three responses.

The degree of impact (see Table 9-13 and Figure 9-26) is significant with a mean of 4.5 and 16.7% indicated the impact was less than significant. There were 44.5% of the respondents who designated five or above on the scale that would indicate the degree of impact was not as significant for this parameter as others. For those who considered "Reliability", one considered the degree of impact to be very significant and no one considered it not significant.

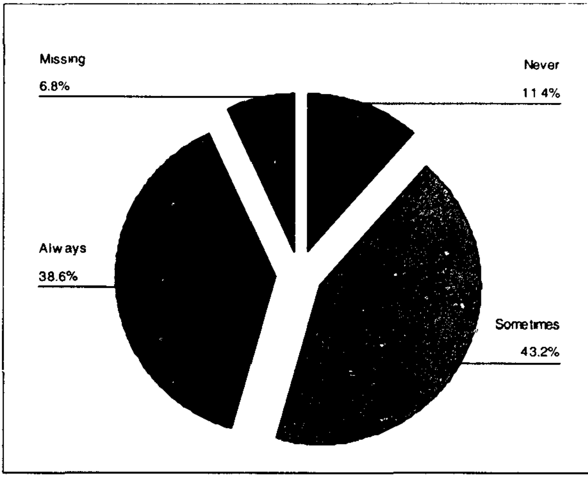

Figure 9-25 Consideration of Reliability

	Degree of Impact	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	2	4.5	5.6	5.6
	3	4	9.1	11.1	16.7
	4	14	31.8	38.9	55.6
	5	6	13.6	16.7	72.2
	6	9	20.5	25.0	97.2
	7	1	2.3	2.8	100.0
Total		36	81.8	100.0	
System Missing		8	18.2		
Total		44	100.0		

Table 9-13 Degree of impact - Reliability

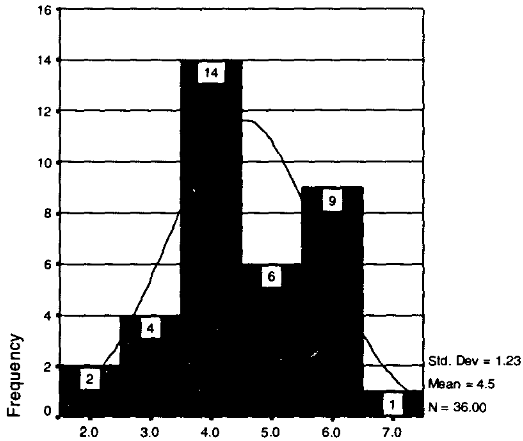

Figure 9-26 Degree of impact - Reliability

9.14 Project Risk

For the parameter "Project Risk", Figure 9-27 shows that this parameter is never considered by 9.1% of the respondents. It is sometimes considered by 38.6% and always considered by 50%. There is 2.3% missing which equates to one response. The degree of impact (see Table 9-14 and Figure 9-28) is very significant with a mean of 5 and 10.5% indicated the impact was less than significant. There were 63.2% of the respondents who designated five or above on the scale which would indicate the degree of impact was more significant for this parameter as others. For those who considered "Project Risk", three considered the degree of impact to be very significant and no one considered it not significant.

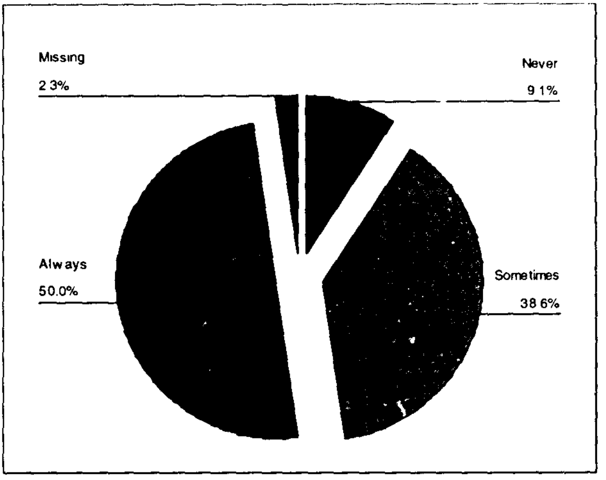

Figure 9-27 Consideration of Project risk

	Degree of Impact	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	1	2.3	2.6	2.6
	3	3	6.8	7.9	10.5
	4	10	22.7	26.3	36.8
	5	8	18.2	21.1	57.9
	6	13	29.5	34.2	92.1
	7	3	6.8	7.9	100.0
Total		38	86.4	100.0	
System Missing		6	13.6		
Total		44	100.0		

Table 9-14 Degree of impact - Project risk

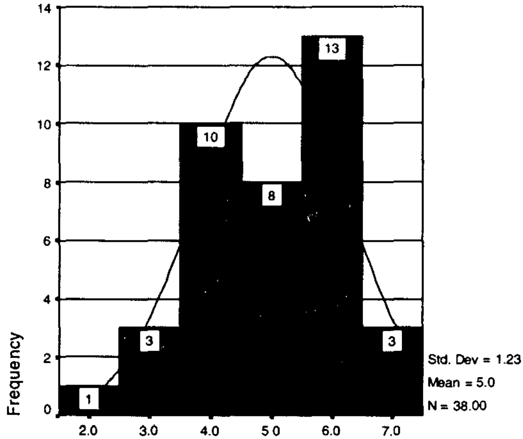

Figure 9-28 Degree of impact - Project risk

9.15 Development Environment

For the parameter "Development Environment", Figure 9-29 shows that this parameter is never considered by 9.1% of the respondents. It is sometimes considered by 36.4% and always considered by 47.7%. There is 6.8% missing which equates to three responses.

The degree of impact (see Table 9-15 and Figure 9-30) is significant with a mean of 4.7 although 11.1% indicated the impact was less than significant. There were 44.5% of the respondents who designated five or above on the scale which would indicate the degree of impact was of significance. For those who considered

"Development Environment", two considered the degree of impact to be very significant and no one considered it not significant.

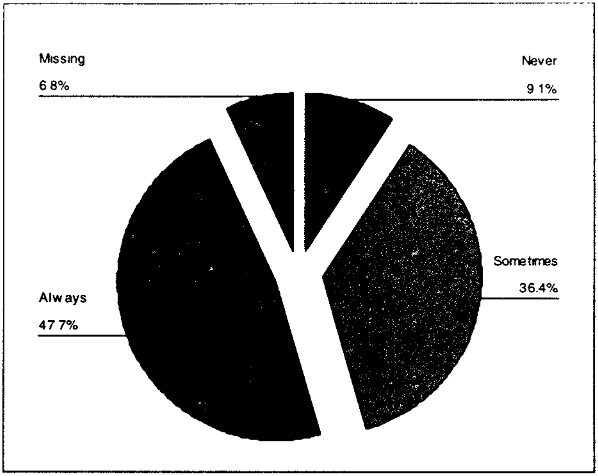

Figure 9-29 Consideration of Development environment

	Degree of Impact	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	1	2.3	2.8	2.8
	3	3	6.8	8.3	11.1
	4	16	36.4	44.4	55.6
	5	5	11.4	13.9	69.4
	6	9	20.5	25.0	94.4
	7	2	4.5	5.6	100.0
Total		36	81.8	100.0	
System Missing		8	18.2		
Total		44	100.0		

Table 9-15 Degree of impact - Development environment

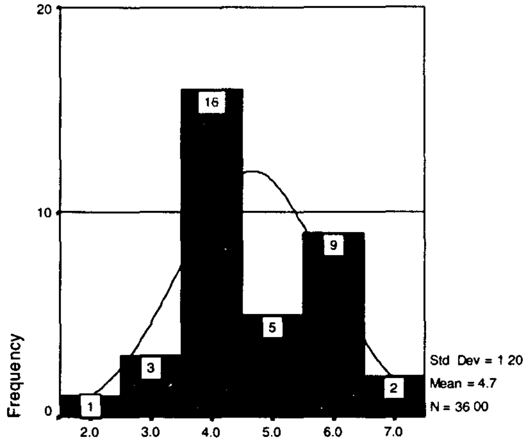

Figure 9-30 Degree of impact - Development Environment

9.16 Schedule Constraints

For the parameter "Schedule Constraints", Figure 9-31 shows that this parameter is never considered by 2.3% or one of the respondents. It is sometimes considered by 36.4% and always considered by 56.8% percent. There is 4.5% missing which equates to two responses.

The degree of impact (see Table 9-16 and Figure 9-32) is very significant with a mean of 5 although 15% indicated the impact was less than significant. There were 57.5% of the respondents who designated five or above on the scale which would indicate the degree of impact was significant. For those who considered "Schedule

Constraints", seven considered the degree of impact to be very significant and no one considered it not significant.

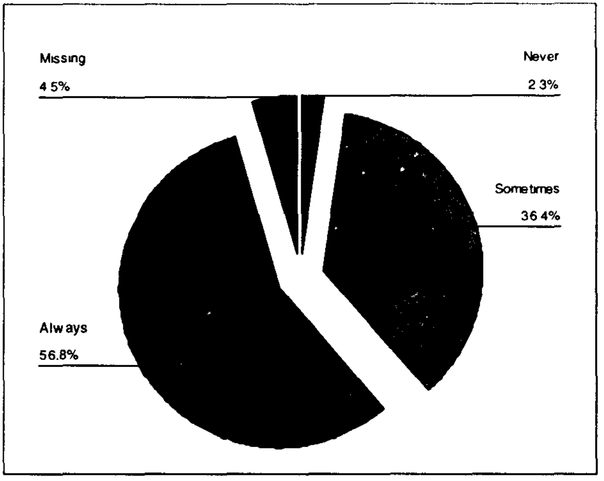

Figure 9-31 Consideration of Schedule constraints

	Degree of Impact	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	3	6	13.6	15.0	15.0
	4	11	25.0	27.5	42.5
	5	7	15.9	17.5	60.0
	6	9	20.5	22.5	82.5
	7	7	15.9	17.5	100.0
Total		40	90.9	100.0	
System Missing		4	9.1		
Total		44	100.0		

Table 9-16 Degree of impact - Schedule constraints

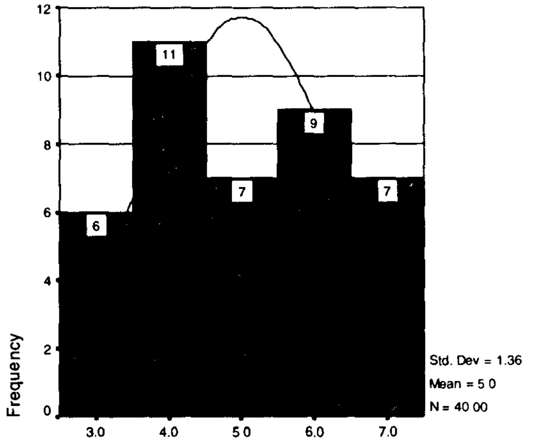

Figure 9-32 Degree of impact - Schedule constraints

9.17 No of Users

For the parameter "No of Users", Figure 9-33 shows that this parameter is never considered by 9.1% of the respondents. It is sometimes considered by 52.3% and always considered by 34.1%. There is 4.5% missing which equates to two responses. The degree of impact (see Table 9-17 and Figure 9-34) is relatively insignificant with a mean of 4 and 39.5% also indicated the impact was less than significant. There were 29% of the respondents who designated five or above on the scale. This is the lowest score for any parameter and would indicate the degree of impact was of the least significance. For those who considered "No of Users", three considered the degree of impact to be very significant and two considered it not significant.

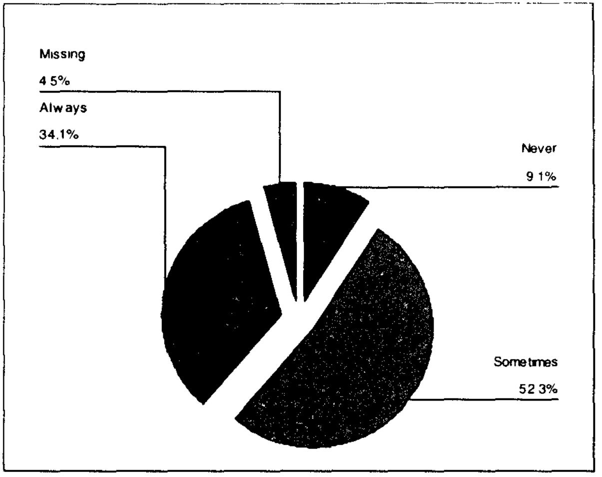

Figure 9-33 Consideration of No of users

	Degree of Impact	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	2	4.5	5.3	5.3
	2	5	11.4	13.2	18.4
	3	8	18.2	21.1	39.5
	4	12	27.3	31.6	71.1
	5	2	4.5	5.3	76.3
	6	6	13.6	15.8	92.1
	7	3	6.8	7.9	100.0
Total		38	86.4	100.0	
System Missing		6	13.6		
Total		44	100.0		

Table 9-17 Degree of impact - No of users

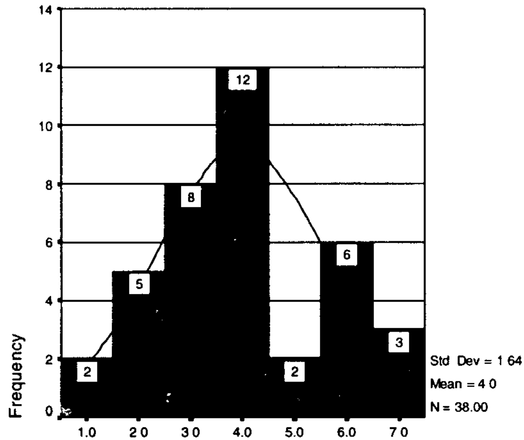

Figure 9-34 Degree of impact - No of users

9.18 Other

An "Other" category was available for the respondents to indicate parameters that were not listed. For the parameter "Other", Figure 9-35 shows that this parameter is sometimes considered by four point five percent and always considered by nine point one percent. There is not a "Never" category and 86.4% are missing.

Some of the additional parameters listed by the respondents such as client risk, loss of key staff, vendor reliability could be considered project risks. Some of the other parameters listed were:

- Country where the project is to installed;
- Degree of change to business processes;

- New technology
- Politics and management support;
- Number of interfaces - data and functional.

The degree of impact (see Table 9-18 and Figure 9-36) is very high with a mean of 5.5 and no one indicated the impact was less than significant. All who responded to this question considered the degree of impact to be significant with none indicating the degree of impact to be less than significant. One considered the degree of impact to be very significant.

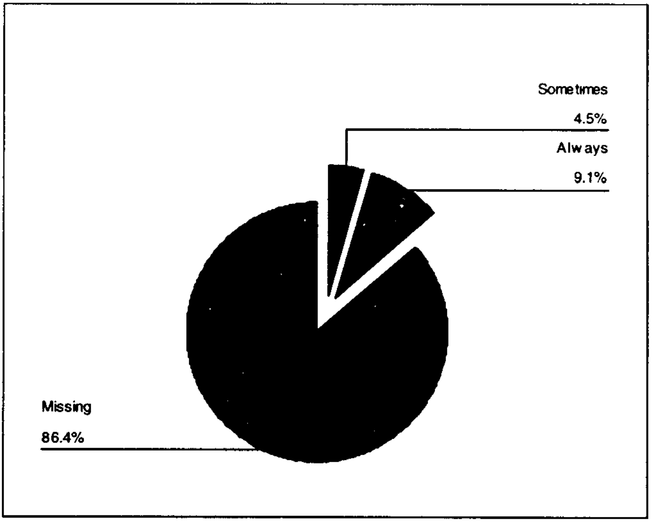

Figure 9-35 Consideration of Other

	Degree of Impact	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	4	1	2.3	16.7	16.7
	5	2	4.5	33.3	50.0
	6	2	4.5	33.3	83.3
	7	1	2.3	16.7	100.0
Total		6	13.6	100.0	
System Missing		38	86.4		
Total		44	100.0		

Table 9-18 Degree of impact - Other

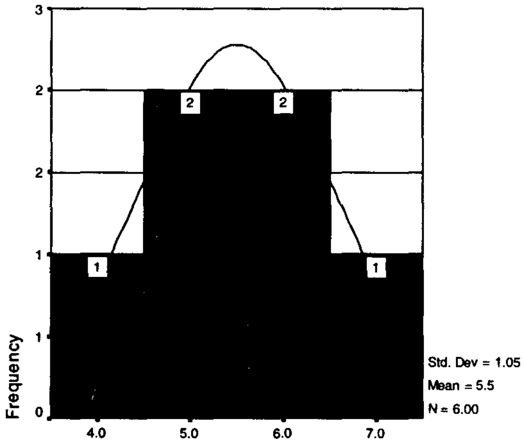

Figure 9-36 Degree of impact - Other

9.19 Conclusions

Table 9-19 gives a summary of some of the significant data in relation to the parameters that would cause nominal project estimates to be modified and their impact in terms of significance. The parameters listed would appear to be a reasonable set as few respondents (six) listed additional ones.

Parameter	Impact Mean	% Above Significant	Standard Deviation	% Who Always Consider	% Who Never Consider
People skills	5.6	70.7	1.3	50.0	4.5
Knowledge of the appln domain	5.3	72.5	1.35	61.4	4.5
Complexity of the problem	5.2	79.4	1.31	63.6	4.5
Algorithmic complexity	4.0	30.3	1.21	15.9	20.5
Stability of the target platform	4.4	44.4	1.64	20.5	15.9
User support	4.8	56.1	1.21	31.8	11.4
Training	4.3	42.2	1.36	31.8	9.1
Maintainability	4.1	39.4	1.39	34.1	6.8
Performance requirements	4.7	45.0	1.12	36.4	4.5
Useability	4.6	48.6	1.42	40.9	9.1
Database size	4.0	31.6	1.56	36.4	11.4
Language to be used	4.3	41.6	1.54	43.2	11.4
Reliability	4.5	44.5	1.23	38.6	11.4
Project risk	5.0	63.2	1.23	50.0	9.1
Development environment	4.7	44.5	1.20	47.7	9.1
Schedule constraints	5.0	57.5	1.36	56.8	2.3
No of users	4.0	29.0	1.64	34.1	9.1
Other	5.5	83.3	1.05	9.1	86.4

Table 9-19 Summary of parameters considered & the degree of impact

All have a mean that is greater than four (four represented a significant degree of impact) and on five parameters one person indicated they were not significant and on three parameters two people indicated it was not significant.

The parameters that appear to be the most important are:

- People skills
- Knowledge of the application domain
- Complexity of the problem
- Project risk and
- Schedule constraints.

These all have a degree of impact mean greater than or equal to five and are always considered by a significant percentage of the respondents.

The degree of impact of "Algorithmic Complexity", "Database Size", and "Number of Users" are less significant that all the other parameters.

The "Algorithmic Complexity" and "Stability of the Target Platform" parameters tend to relate more to the construction of real time systems and hence due to most projects not being in this application domain the percentage of people who never consider these is relatively high.

Chapter 10 Estimating Practices: WA Current Practice

In the survey questionnaire, questions were asked regarding the practices discarded and practices currently used. Details of the specific techniques were sought and also the reasons for the techniques being used or no longer being used. The responses elicited were in part not complete. Especially those relating to the specifics of the techniques and practices employed.

Specific questions relating to the respondents normal practices to estimate size, duration, effort and cost that contained different categories and seven point scales to determine how the respondents used them and their views of the accuracy, ease of use were more successful in obtaining useful data.

10.1 Discarded Practices

Questions were asked relating to techniques discarded in order to obtain information that may have revealed weaknesses in them. However, the responses indicated that only seven respondents had discarded some techniques and the answers gained contained insufficient detail for analysis. Thirty-five advised they had not discarded any techniques. Three responses indicated Function Point Analysis was no longer used for the following reasons:

- as it was found to be inaccurate,
- too time consuming and
- did not suit modern technology.

Figure 10-1 summaries the responses.

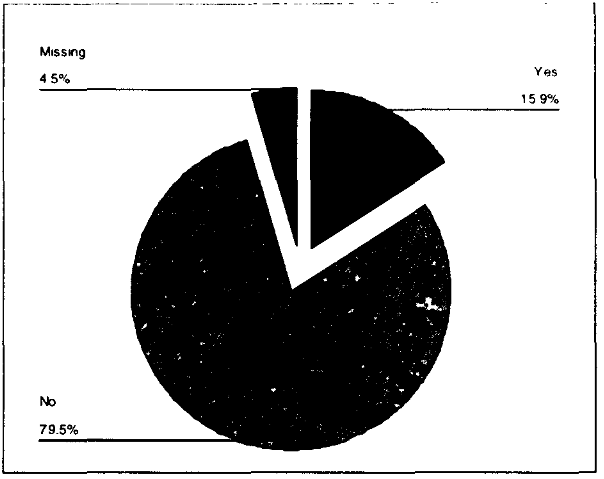

Figure 10-1 Techniques discarded

10.2 Current practices

As can be seen from Table 10-1 and Figure 10-2 analogy, expert judgement and work breakdown are the most widely used estimating techniques. Figure 10-2 graphs the techniques used only by the respondents, those used by the organisation and those used by both the respondent and the organisation. The numbers had been added together to give a combined view. The responses relating to "not used" are also contained in the graph. The most prevalent technique used by an individual is analogy, for an organisation expert judgement and for both work breakdown. Algorithmic techniques tend not to be widely used and have the highest scores in the "not used" category. It is interesting to note analogy, expert judgement and work breakdown are the most frequently used techniques with work breakdown being the highest.

		Analogy		Expert Judgement		Work Breakdown		Algorithmic Model	
	Use of Techniques	Frequency	Percent	Frequency	Percent	Frequency	Percent	Frequency	Percent
Valid	Self	15	34.1	10	22.7	9	20.5	4	9.1
	Organisation	1	2.3	5	11.4	3	6.8	0	0
	Both	16	36.4	16	36.4	25	56.8	7	15.5
	Combined	32	72.8	31	70.5	37	84.1	11	24.6
	Not used	7	15.9	9	20.5	2	4.5	29	65.9
System Missing		5	11.4	4	9.1	5	11.4	4	9.1
Total		44	100.0	44	100.0	44	100.0	44	100.0

Table 10-1 Technique usage

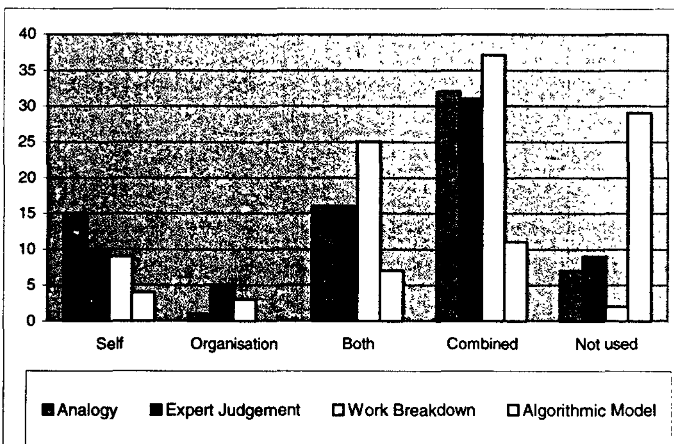

Figure 10-2 Technique Usage

The data in Table 10-1 also raises another issue worthy of note. That is, one of the initial questions asked in the questionnaire was "What techniques the respondents had used in their career" and requested a rating of their usefulness. The responses indicated that twenty-two or fifty percent of the respondents had used LOC techniques and twenty-three or 52.3% had used Function Point analysis. See Tables 7-12 and 7-13. This information when combined with the data from Table 10-1, it can be seen that only approximately half of the respondents who have used algorithmic techniques are still using them.

	Training Readily Available	Found to be Accurate	Easy to Use	Organisational Standard	Appropriate for the Projects
Analogy	0	10	21	3	21
Expert Judgement	1	16	22	4	24
Work Breakdown	5	26	25	12	32
Algorithmic Model	2	3	3	3	8

Table 10-2 Reasons for Technique use

Figure 10-3 provides a stacked bar graph on the reasons why the particular techniques are used. The principle reasons being they are appropriate for the projects, easy to use and found to be accurate. The availability of training does not appear to affect the usage of techniques although there is a correlation with training availability and the responses to "found to be accurate" and "organisational standard". It would appear many organisations do not mandate a technique and where they do set a standard, it is principally based on a work breakdown structure.

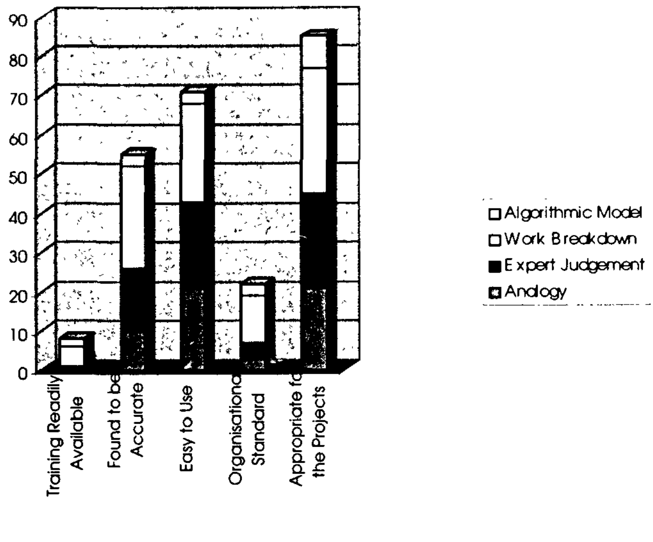

Figure 10-3 Reasons for technique usage

The respondent's normal practices relating to their manner for estimating the different dimensions of a project (size, duration, effort and cost) were gauged. They were asked if these different dimensions were estimated and if so, what techniques were used, how often were they used, how easy were they to use and their degree of accuracy. Table 10-3 shows that the respondents nearly always estimated size and duration with 95.5% and 97.7% doing so respectively. Effort and cost were estimated to a lesser extent with 79.5% and 84.1% doing so respectively. Note there was only one missing response in the data and this was for estimating project effort.

	Size	Duration	Effort	Cost
Estimated	95.5%	97.7%	79.5%	84.1%
Not Estimated	4.5%	2.3%	18.2%	15.9%

Table 10-3 Project dimensions estimated

This response raises a fundamental question in that as can be clearly seen 95.5% estimate project size, however, only eleven or 25% of the respondents are using algorithmic models. Algorithmic models being the only techniques that incorporate some fundamental size measure such as function points or lines of code. This would be an interesting area to explore in future research as to what the respondents size measures and concepts are. (Also, what is size in relation to software)?

10.3 Size Estimating Practices

The size statistics relating to frequency of use are detailed in Table 10-4 and Figures 10-4, 5, 6, 7 and 8. The data correlates well with the data obtained through another question relating to technique usage - see Table 10-1. This gives a degree of confidence in the internal consistency of the responses.

	Number of Responses			Statistic's		
	Valid	%	Missing	Mean	Median	Mode
Analogy	32	72.7	12	4.81	5.00	4
Expert judgement	36	81.8	8	5.64	6.00	6
Work breakdown	37	84.1	7	5.78	6.00	6
Algorithmic model	13	29.5	31	3.92	4.00	4
Other	5	11.4	39	3.80	4.00	1*

* Multiple modes exist. The smallest value is shown.

Table 10-4 Size Statistics -Frequency of use

Work breakdown had the greatest percentage of use, for estimating size at 84.1% closely followed by expert judgement at 81.8% and analogy at 72.7%. Algorithmic models were only used by 29.5% of the respondents. The frequency with which the techniques were used showed that work breakdown and expert judgement tend to be always used with means of 5.78 and 5.64 respectively Both medians and modes were six which provides further confirmation that these two techniques tend to be always used. Algorithmic models on the other hand only tend to be used sometimes with a mean of 3.92 and a median and mode of four. The histogram in Figure 10-7 also clearly shows this.

Five respondents indicated they use other techniques for estimating. The techniques given could be classified as sub-sets or derivatives of the main categories given. Eg two additional techniques given were RAD/Function Points - an extension of FPA and Experts workshop - an extension of expert judgement. These techniques, in general were only sometimes used.

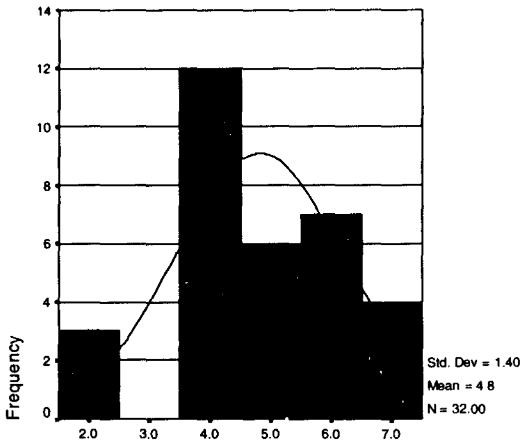

Figure 10-4 Size - Analogy Frequency of use

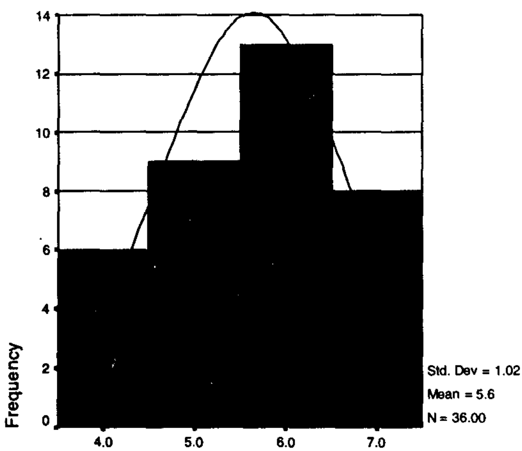

Figure 10-5 Size - Expert Judgement Frequency of use

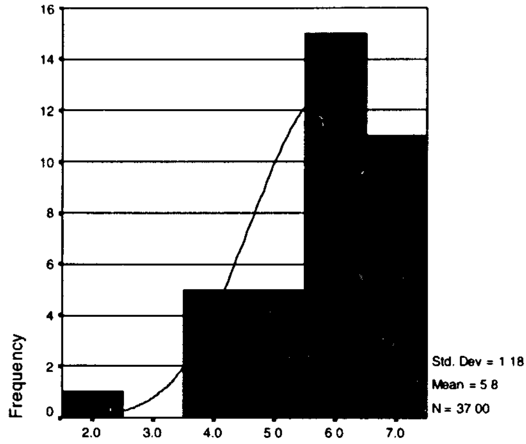

Figure 10-6 Size - Work Breakdown Frequency of use

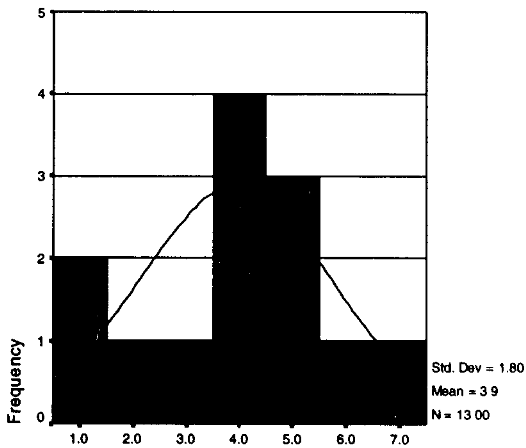

Figure 10-7 Size - Algorithmic model Frequency of use

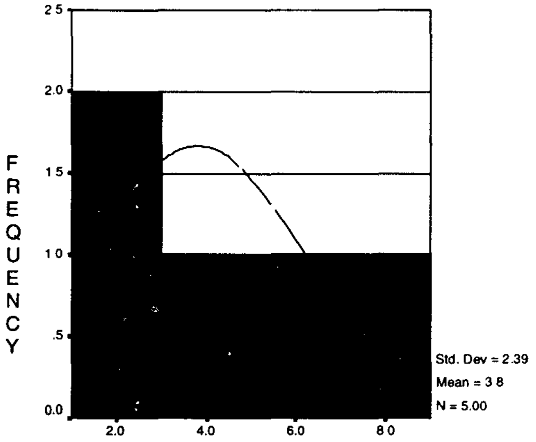

Figure 10-8 Size - Other Frequency of use

The size statistics relating to ease of use are detailed in Table 10-5 and Figures 10-9, 10, 11,12 and 13.

	Number of Responses			Statistic's		
	Valid	%	Missing	Mean	Median	Mode
Analogy	29	65.9	15	4.62	5.00	4
Expert judgement	34	77.3	10	5.15	5.00	5
Work breakdown	34	77.3	10	4.82	5.00	5
Algorithmic model	11	25.0	33	4.27	4.00	4*
Other	5	11.4	39	3.80	4.00	1*

* Multiple modes exist. The smallest value is shown.

Table 10-5 Size Statistics - Ease of use

Expert judgement and work breakdown are reported by 77.3% of the respondents as the easiest to use with means of 5.15 and 4.82 respectively. This indicates they had above average ease of use and this is further supported by the medians and modes being five. The people who use analogy, 65.9%, and algorithmic models, 25%, also indicated their ease of use was above average. The respondents who used other techniques indicated their ease of use tended to be below average. The histograms reveal the ease of use of analogy, expert judgement and work breakdown to be clustered towards the excellent side of the scale whereas the responses regarding the algorithmic models are more ambivalent with responses across the scale.

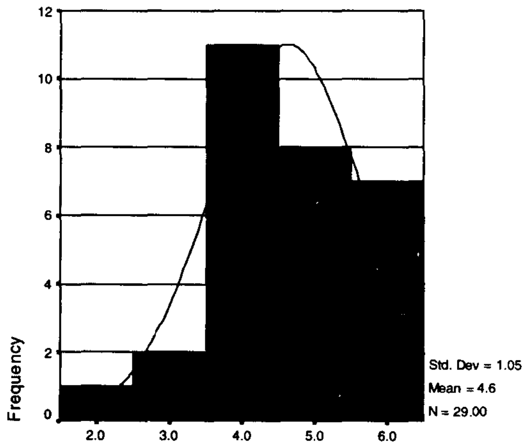

Figure 10-9 Size - Analogy ease of use

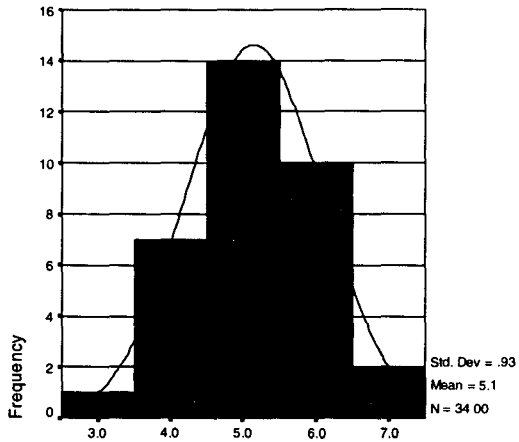

Figure 10-10 Size - Expert judgement ease of use

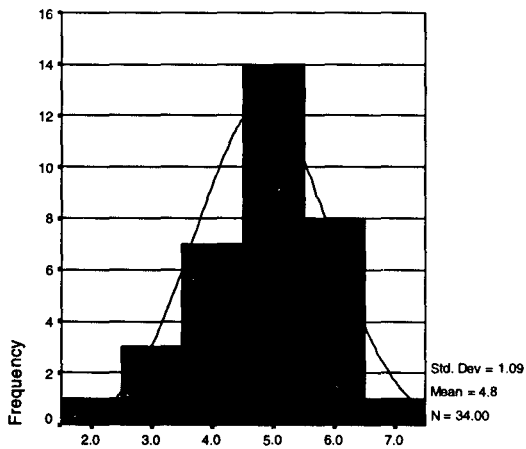

Figure 10-11 Size - Work breakdown ease of use

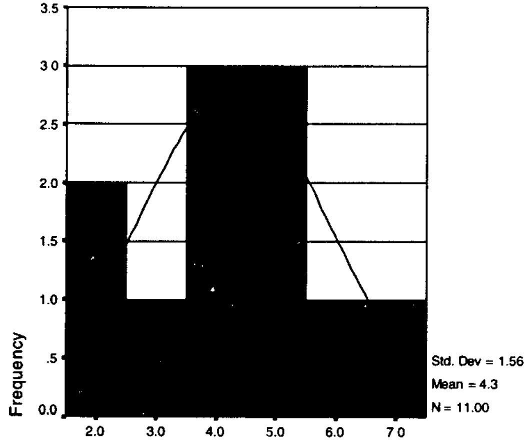

Figure 10-12 Size Algorithmic model ease of use

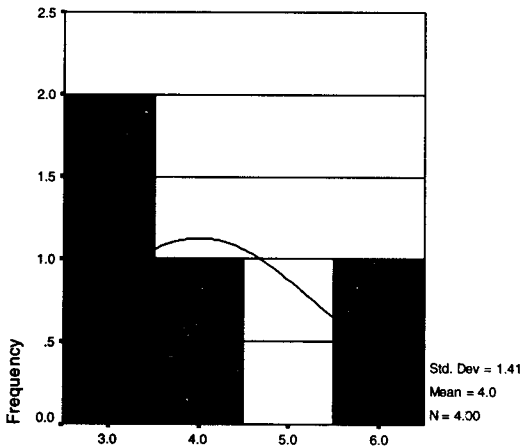

Figure 10-13 Size other ease of use

The size statistics relating to accuracy are detailed in Table 10-6 and Figures 10-14, 15, 16, 17 and 18.

	Number of Responses			Statistic's		
	Valid	%	Missing	Mean	Median	Mode
Analogy	30	68.1	14	4.27	4.00	4
Expert judgement	36	81.1	9	4.60	5.00	5
Work breakdown	36	81.1	8	5.08	5.00	6
Algorithmic model	11	25.0	33	4.36	5.00	5
Other	4	9.1	40	5.00	5.00	5

* Multiple modes exist. The smallest value is shown.

Table 10-6 Size statistics - Accuracy

Work breakdown, reported by 81.1%, is the most accurate sizing technique with a mean of 5.08, a median of five and a mode of six. The other techniques are similar in their accuracy although analogy would appear to be less accurate that the others.

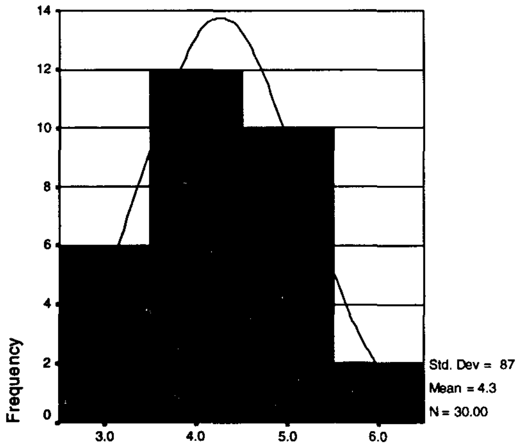

Figure 10-14 Size - Analogy accuracy

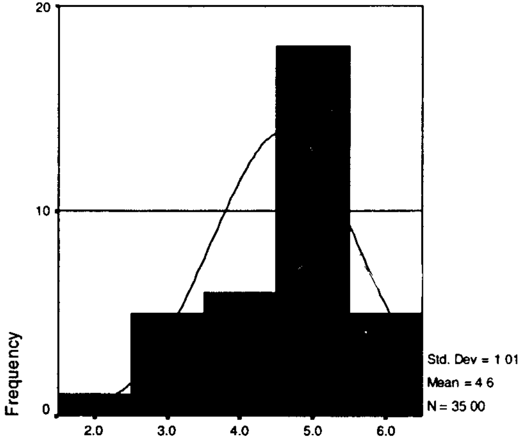

Figure 10-15 Size - Expert judgement accuracy

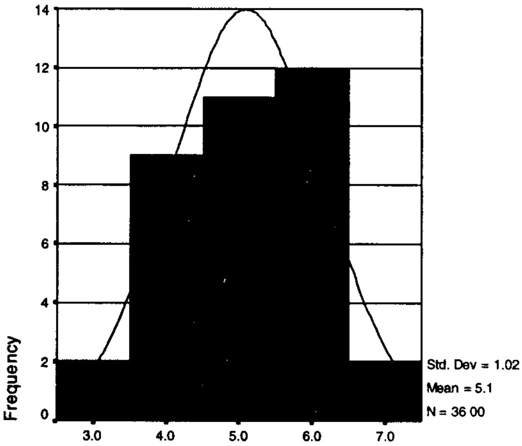

Figure 10-16 Size - Work breakdown Accuracy

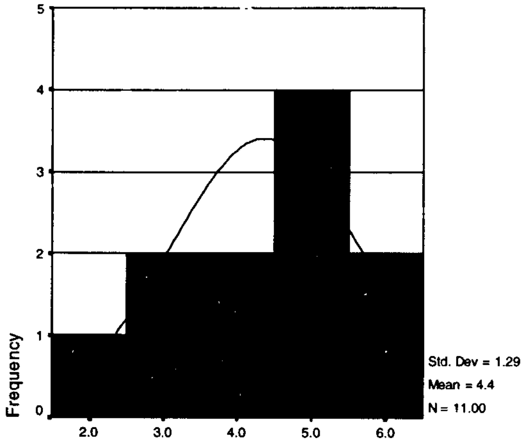

Figure 10-17 Size Algorithmic model accuracy

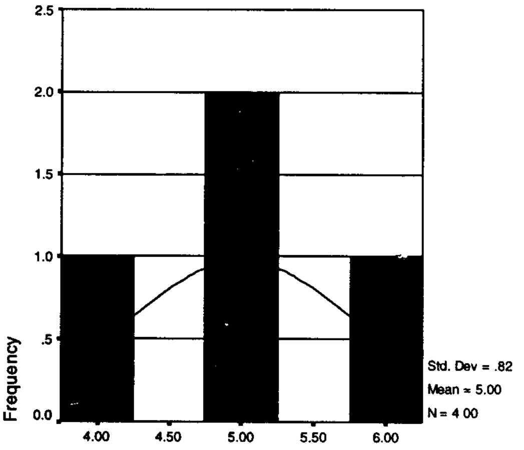

Figure 10-18 Size - Other accuracy

10.4 Duration Estimating Practices

The duration statistics relating to frequency of use are detailed in Table 10-7.

	Number of Responses			Statistic's		
	Valid	%	Missing	Mean	Median	Mode
Analogy	27	61.4	17	5.19	5	6
Expert judgement	33	75.0	11	5.55	6	6
Work breakdown	40	91.1	4	5.63	6	6
Algorithmic model	9	20.5	35	4.11	4	4*
Other	5	11.4	39	4.20	5	1*

* Multiple modes exist. The smallest value is shown.

Table 10-7 Duration Statistics - Frequency of use

Work breakdown had the greatest percentage of use, for estimating duration at 91.1%, representing forty respondents, followed by expert judgement at 75% and analogy at 61.4%. Algorithmic models were only used by 20.5% of the respondents. The frequency with which the techniques were used showed that work breakdown and expert judgement tend to be always used with means of 5.53 and 5.55 respectively. Both medians and modes were six which provides further confirmation that these two techniques tend to be always used. Algorithmic models on the other hand are only used by nine respondents (20.5%) and they only tend to use the techniques sometimes as demonstrated with a mean of 4.11 and a median of four and modes of four and five.

The five respondents who indicated they use other techniques for estimating duration in general said they were only sometimes used.

The data clearly shows that the most popular technique is work breakdown, with the greatest percentage of people indicating they use it, and that they tend to use it most frequently.

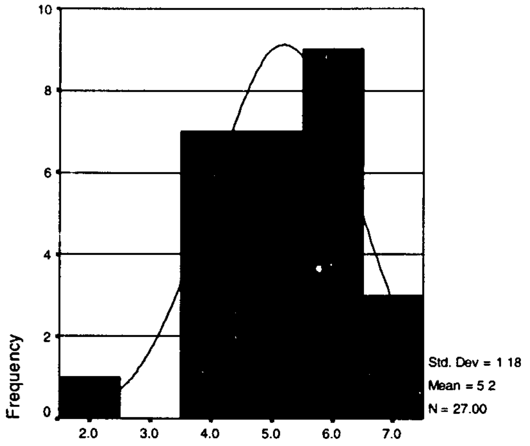

Figure 10-19 Duration - Analogy Frequency of use

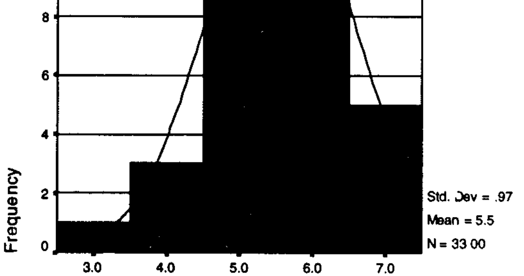

Figure 10-20 Duration - Expert judgement Frequency of use

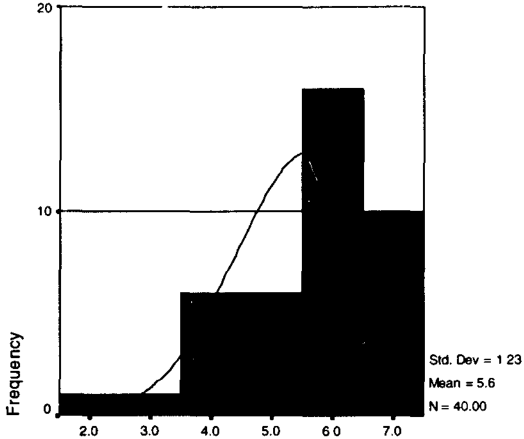

Figure 10-21 Duration Work Breakdown Frequency of use

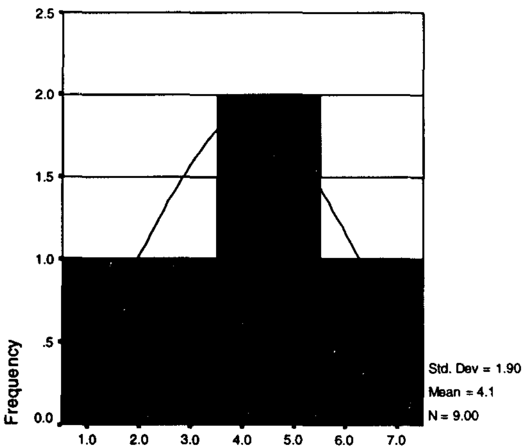

Figure 10-22 Duration - Algorithmic model Frequency of use

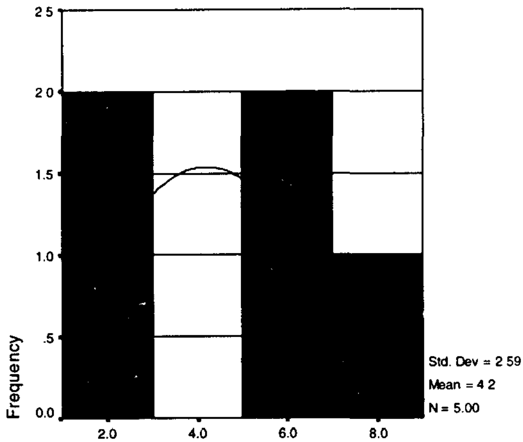

Figure 10-23 Duration - Other Frequency of use

The duration statistics relating to ease of use are detailed in Table 10-8 and Figures 10-24, 25,26 and 27.

	Number of Responses			Statistic's		
	Valid	%	Missing	Mean	Median	Mode
Analogy	25	56.8	19	4.76	5	4
Expert judgement	31	70.5	13	5.26	5	5
Work breakdown	37	84.1	7	4.86	5	4
Algorithmic model	7	15.9	37	4.71	5	3*
Other	3	6.8	41	4.33	5	5

* Multiple modes exist. The smallest value is shown.

Table 10-8 Duration Statistics - Ease of use

Work breakdown had the greatest percentage of respondents at 84.1% willing to provide an opinion on the ease of use of any of the techniques. This would suggest that this is the technique with which they are most familiar and hence able to offer an opinion. Although they rated its ease of use slightly less than the expert judgement technique with a mean of 4.86, a median of 5 and a mode of four. Expert judgement had 70.5% offer an opinion which showed this was the easiest technique to use with a mean of 5.26 and a median and mode of five. Analogy was used by 56.8% who indicated that this technique was also easy to use. Only seven respondents (15.9%) gave an opinion on algorithmic models of which indicated in general the technique was above average in easy to use although three indicated average to below average ease of use. Only three people gave opinions on other techniques and hence a graph of "other" does not add any information.

The data clearly shows that the most popular technique is work breakdown although expert judgement is considered easier to use.

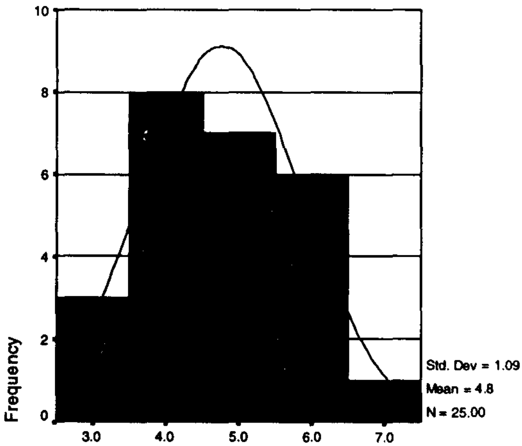

Figure 10-24 Duration - Analogy Ease of use

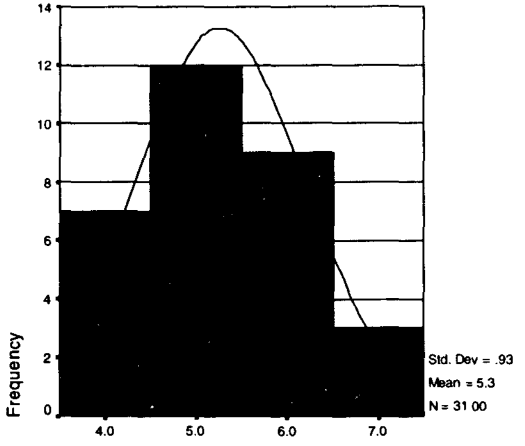

Figure 10-25 Duration - Expert judgement Ease of use

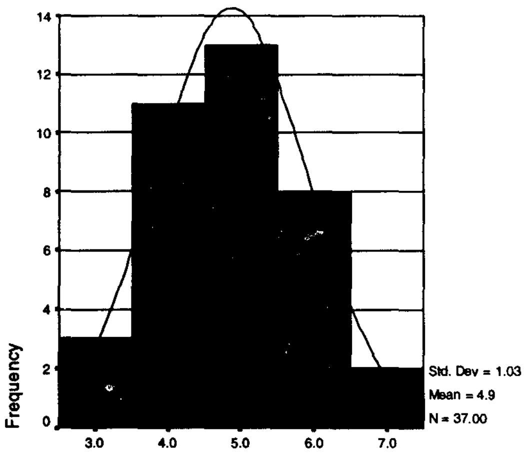

Figure 10-26 Duration - Work breakdown Ease of use

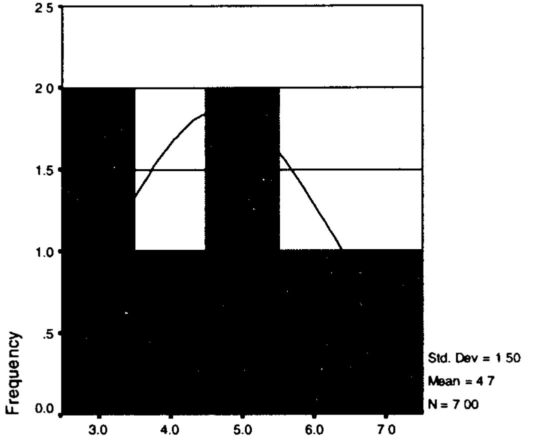

Figure 10-27 Duration - Algorithmic model Ease of use

The duration statistics relating to accuracy are detailed in Table 10-9 and Figures 10-28, 29, 30, and 31.

	Number of Responses			Statistic's		
	Valid	%	Missing	Mean	Median	Mode
Analogy	26	59.1	18	4.42	4	4
Expert judgement	32	72.7	12	4.66	5	5
Work breakdown	39	88.6	5	4.95	5	6
Algorithmic model	8	18.2	36	4.63	5.5	6
Other	4	9.1	40	5	-	5

Table 10-9 Duration Statistics - Accuracy

The data shows the most accurate technique is work breakdown with a mean of 4.95, a median of five and a mode of six. This is, once again, the most popular technique with 88.6% of respondents able to offer an opinion. Expert judgement is the second most accurate with a mean of 4.66 and mode and median of five. The data regarding algorithmic models is interesting as it appears to contain two groups. The histogram, Figure 10-31, shows this with one group advising it is less than accurate and the other indicating it is very accurate. This may be a result of only having eight respondents providing an opinion in this area.

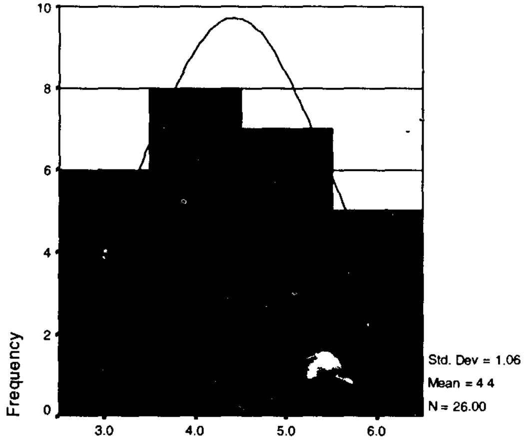

Figure 10-28 Duration - Analogy Accuracy

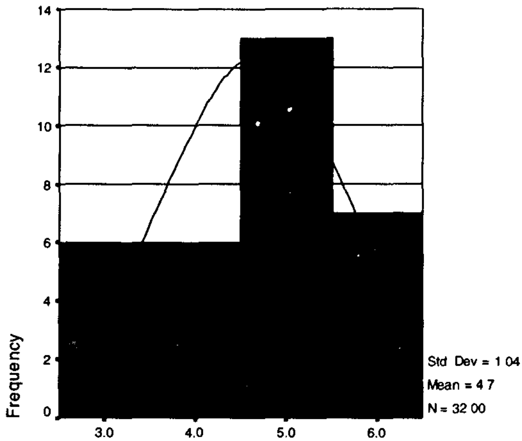

Figure 10-29 Duration - Expert judgement Accuracy

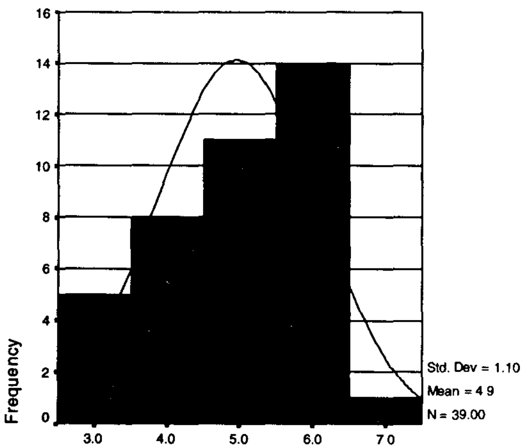

Figure 10-30 Duration - Work breakdown Accuracy

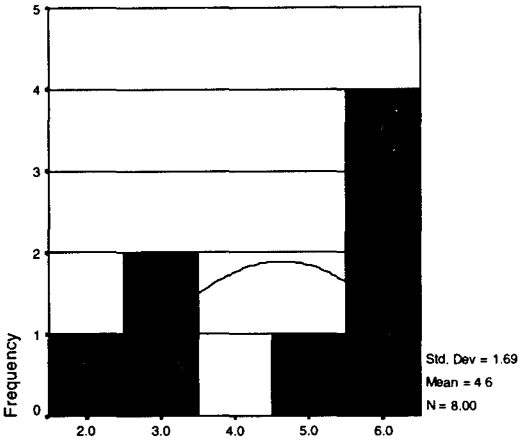

Figure 10-31 Duration - Algorithmic model Accuracy

10.5 Effort Estimating Practices

The effort statistics relating to frequency of use are detailed in Table 10-10 and Figures 10-32, 33, 34, and 35.

	Number of Responses		Statistic's			
	Valid	%	Missing	Mean	Median	Mode
Analogy	22	50.0	22	4.86	5	5
Expert judgement	25	56.8	19	5.28	5	6
Work breakdown	29	65.9	15	5.69	6	6*
Algorithmic model	7	15.9	37	3.86	4	4
Other	3	6.8	41	3.33	2	1*

- Multiple modes exist. The smallest value is shown.

Table 10-10 Effort Statistics - Frequency of use

Work breakdown, expert judgement and analogy are frequently used with work breakdown having the greatest number of respondents (65.9%) providing data. Work breakdown was also indicated to be the most frequently used technique with a mean of 5.69 and having a median of six and modes of six and seven. No-one indicated work breakdown and expert judgement were used less than sometimes. Seven respondents representing 15.9% used algorithmic techniques to estimate effort and the resultant mean of 3.86 and median and mode of four indicated the technique was used sometimes.

Three respondents (6.8%) indicated they infrequently used other techniques for estimating effort. Note a graph is not shown as it adds little value.

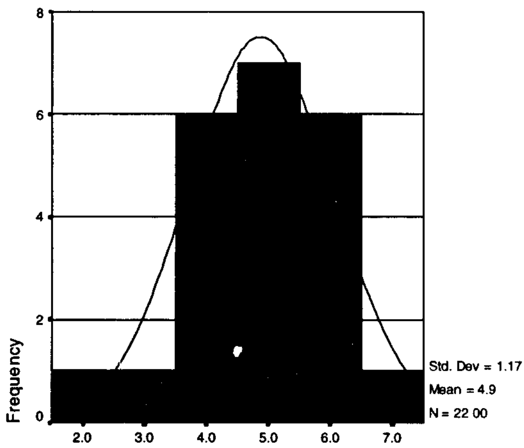

Figure 10-32 Effort - Analogy Frequency of use

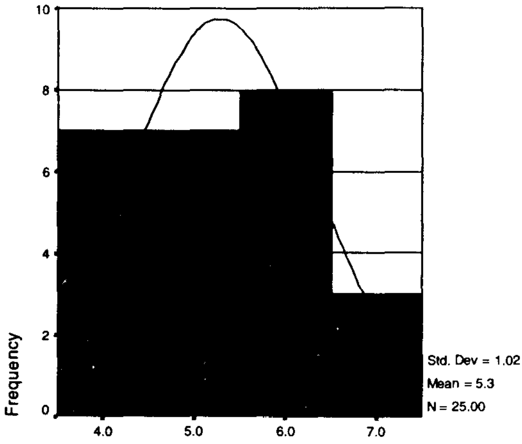

Figure 10-33 Effort - Expert Judgement Frequency of use

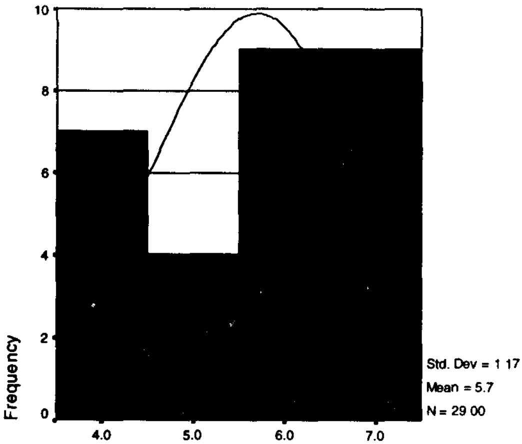

Figure 10-34 Effort - Work Breakdown Frequency of use

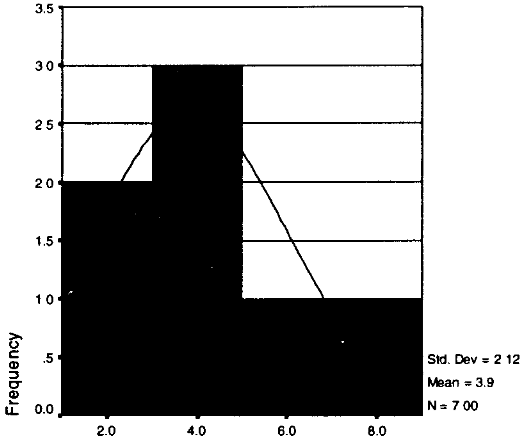

Figure 10-35 Effort - Algorithmic model Frequency of use

The effort statistics relating to ease of use are detailed in Table 10-11 and Figures 10-36, 37, 38 and 39.

	Number of Responses			Statistic's		
	Valid	%	Missing	Mean	Median	Mode
Analogy	21	47.7	23	4.67	5	5
Expert judgement	25	56.8	19	5.16	5	5
Work breakdown	28	63.6	16	4.96	5	4
Algorithmic model	6	13.6	38	4.17	4	4*
Other	2	4.5	42	4.50	4.50	3*

* Multiple modes exist. The smallest value is shown.

Table 10-11 Effort Statistics - Ease of use

Work breakdown is used by the greatest percentage of respondents at 63.6% however, its ease of use is slightly less than the expert judgement technique. Expert judgement is shown to be above average in its ease of use with no responses indicating it was below average and four indicating a rating of excellent.

The 13.6% of the respondents using algorithmic techniques fall into two groups. The majority indicating the ease of use was poor to average and a minority indicating the techniques to be slightly less than excellent.

Of the two respondents using other techniques one indicated it was below average and the other indicated it was above average in ease of use. The graph relating to "other" does not add any value to this work as there are only two respondents in this category.

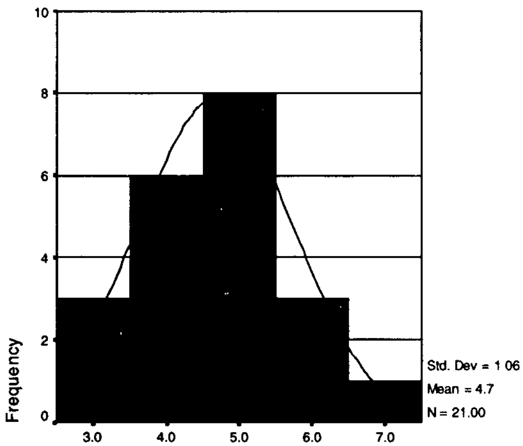

Figure 10-36 Effort - Analogy Ease of use

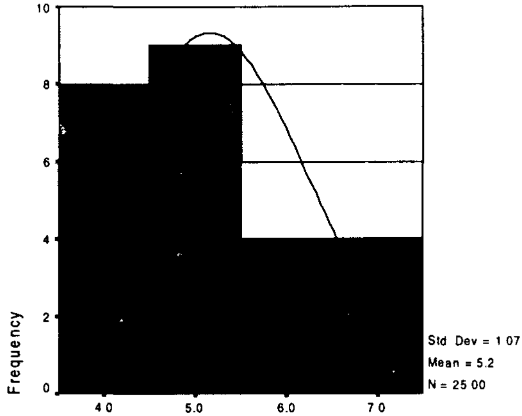

Figure 10-37 Effort - Expert judgement Ease of use

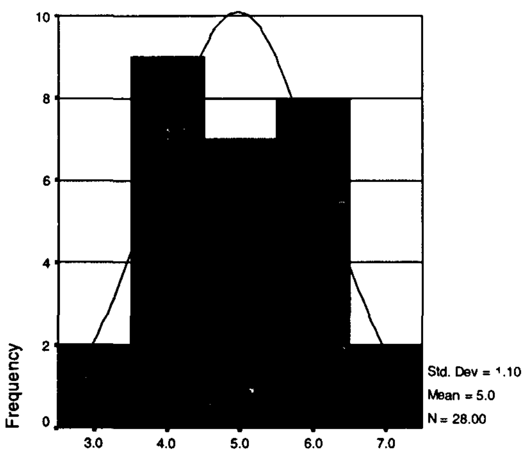

Figure 10-38 Effort - Work breakdown Ease of use

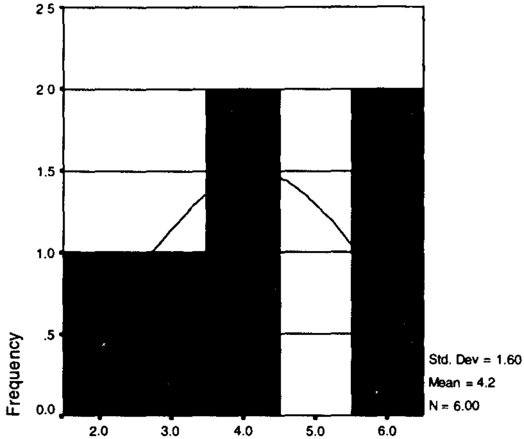

Figure 10-39 Effort - Algorithmic model Ease of use

The effort statistics relating to accuracy are detailed in Table 10-12 and Figures 10-40, 41, 42 and 43. The graph relating to "other" does not add any value as there are only two respondents in this category.

	Number of Responses			Statistic's		
	Valid	%	Missing	Mean	Median	Mode
Analogy	21	47.7	23	4.14	4	4
Expert judgement	25	56.8	19	4.68	5	5
Work breakdown	29	65.9	15	5.00	5	4*
Algorithmic model	6	13.6	38	5.17	5.50	6
Other	2	4.5	42	5.50	5.50	5*

* Multiple modes exist. The smallest value is shown.

Table 10-12 Effort Statistics - Accuracy

On balance it could be said that the most accurate technique was work breakdown although its mean at five was less that that indicated for the algorithmic (5.17) and other (5.5) techniques an significant percentage of respondents 65.9% used the technique. Four of which indicated the technique was very accurate.

The 13.6% who use algorithmic techniques indicated the techniques were accurate to less than very accurate. None indicated algorithmic techniques were inaccurate.

The two respondents who use other techniques indicated they were above average in their accuracy.

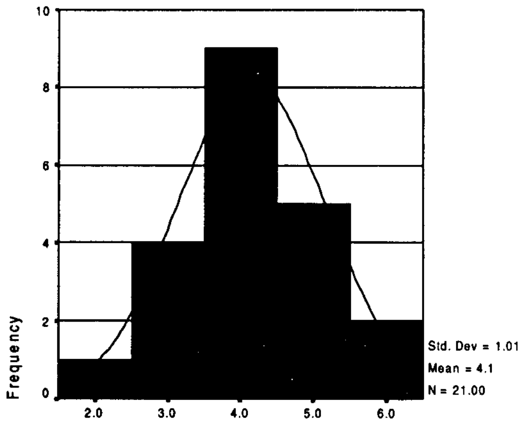

Figure 10-40 Effort - Analogy Accuracy

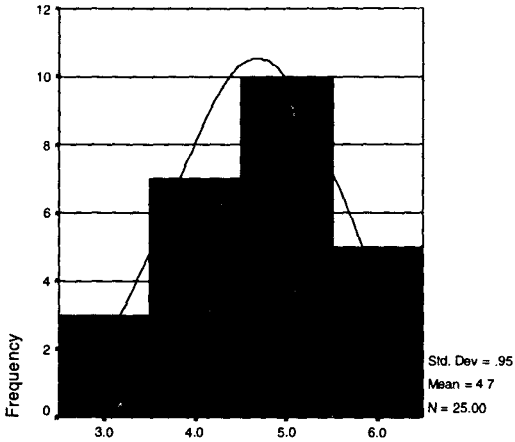

Figure 10-41 Effort - Expert Judgement Accuracy

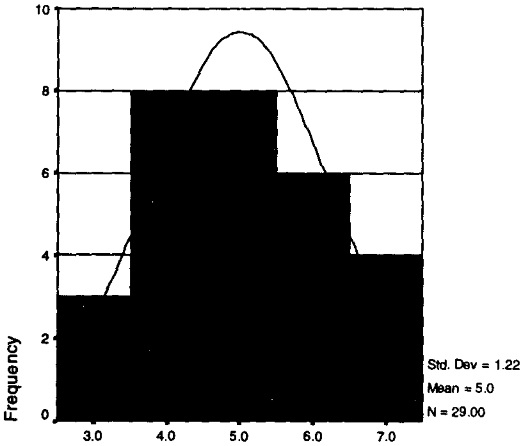

Figure 10-42 Effort - Work breakdown Accuracy

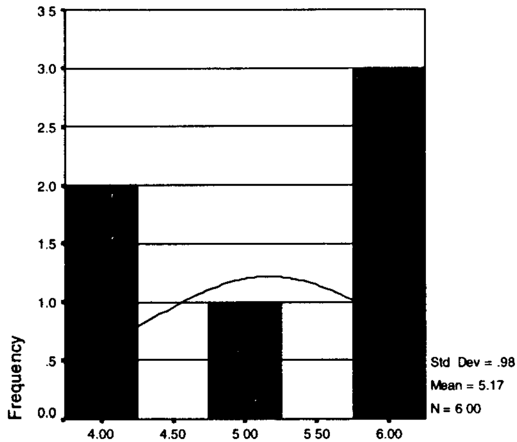

Figure 10-43 Effort - Algorithmic model Accuracy

10.6 Cost Estimating Practices

The cost statistics relating to frequency of use are detailed in Table 10-13 and Figures 10-44, 45, 46 and 47. The graph relating to "other" does not add any value as there are only three respondents in this category.

	Number of Responses			Statistic's		
	Valid	%	Missing	Mean	Median	Mode
Analogy	21	47.7	23	4.71	5	4
Expert judgement	26	59.1	18	5.23	5	6
Work breakdown	35	79.5	9	5.71	6	6
Algorithmic model	9	20.5	35	4.33	4	4
Other	3	6.8	41	3.33	2	1*

* Multiple modes exist. The smallest value is shown.

Table 10-13 Cost Statistics - Frequency of use

Work breakdown is used by the greatest percentage of respondents and also has the greatest frequency of use with a mean of 5.71 and a median and mode of six. Work breakdown also had the highest number of responses, ten, who indicated they always used it. Expert judgement had 59.1% of the respondents who indicated they tended to always use the technique with a mean of 5.53 and a median of five and mode of six. Analogy was not as frequently used as work breakdown and expert judgement with 47.7% responding that the frequency of use was greater than sometimes with a mean of 4.71 and a median of five and a mode of four. Algorithmic techniques had nine responses or 20.5% who indicated it was used sometimes with a mean of 4.33 and a median and mode of four. Algorithmic technique usage was spread across the range of frequencies. The use of other techniques by 6.8% of the respondents tended to be used less frequently than sometimes.

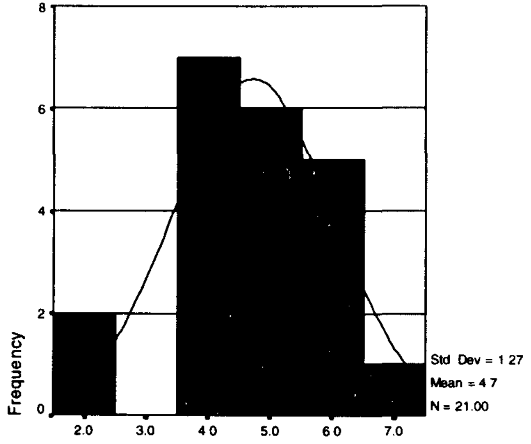

Figure 10-44 Cost - Analogy Frequency of use

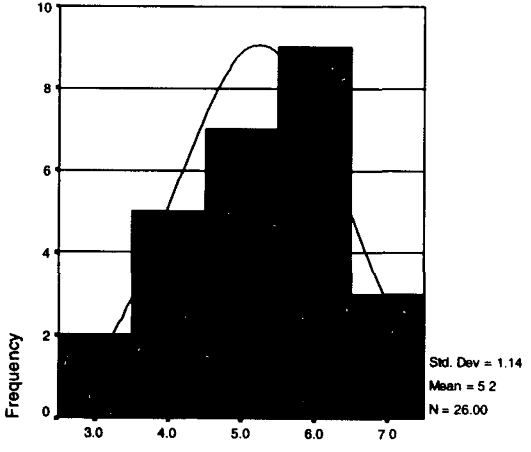

Figure 10-45 Cost - Expert Judgement - Frequency of use

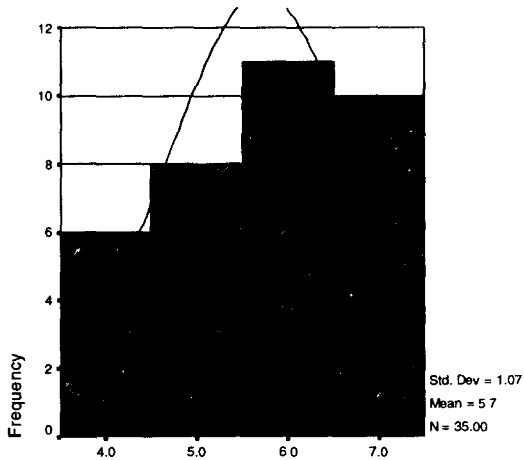

Figure 10-46 Cost - Work Breakdown Frequency of use

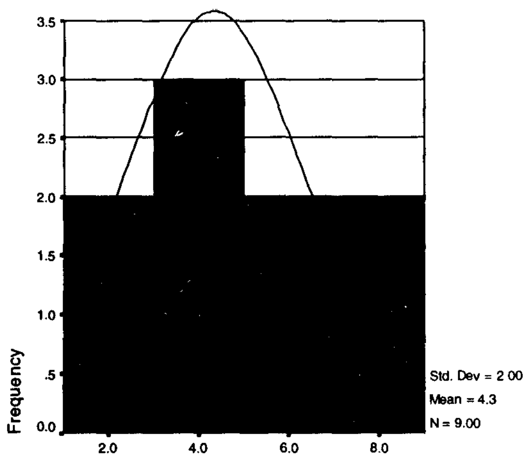

Figure 10-47 Cost - Algorithmic model Frequency of use

The cost statistics relating to ease of use are detailed in Table 10-14 and Figures 10-48, 49, 50 and 51. The graph relating to "other" does not add any value as there are only two respondents in this category.

	Number of Responses			Statistic's		
	Valid	%	Missing	Mean	Median	Mode
Analogy	20	45.5	24	4.60	5	5
Expert judgement	27	61.4	17	5.04	5	5
Work breakdown	34	77.3	10	5.12	5	5
Algorithmic model	6	13.6	38	5.00	5	4*
Other	2	4.5	42	4.50	4.50	3*

* Multiple modes exist. The smallest value is shown.

Table 10-14 Cost Statistics - Ease of use

The bulk of the respondents, 77.3%, have indicated the work breakdown technique is the easiest to use with a mean of 5.12 and a median and mode of five. Expert judgement is deemed the second easiest to use with a mean of 5.04 and a median and mode of five. No one indicated it was below average in its ease of use. Table 10-14 show the mean for algorithmic techniques to be five although when Figure 10-51 is examined two separate views appear to be represented. One indicates the ease of use to be below average to average whilst the other indicates opinions of well above average to excellent.

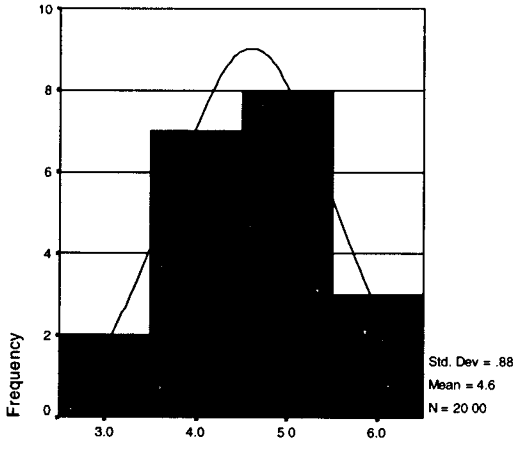

Figure 10-48 Cost - Analogy Ease of use

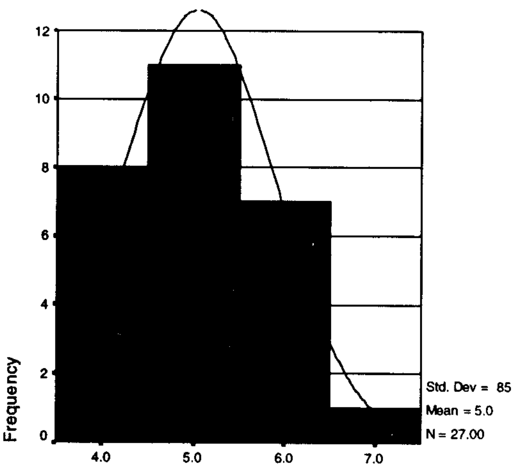

Figure 10-49 Cost - Expert Judgement Ease of use

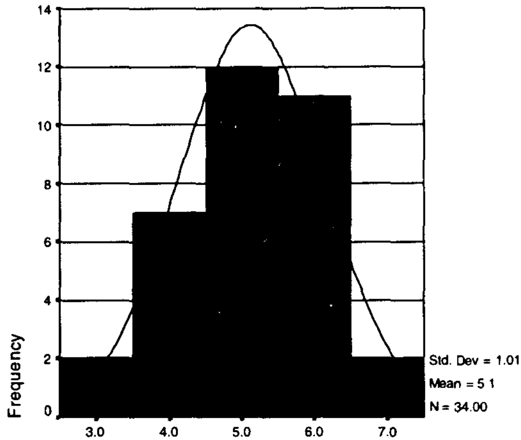

Figure 10-50 Cost - Work breakdown Ease of use

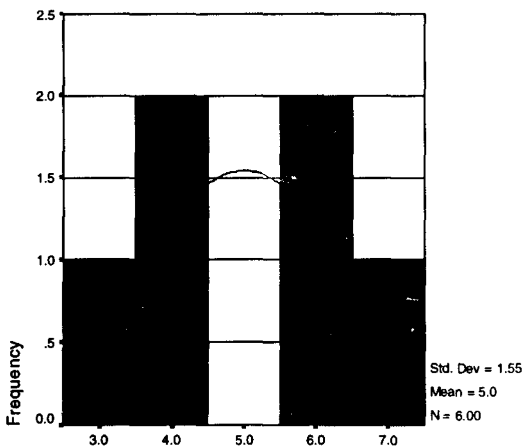

Figure 10-51 Cost - Algorithmic model Ease of use

The cost statistics relating to accuracy are detailed in Table 10-15 and Figures 10-52, 53, 54 and 55. The graph relating to "other" does not add any value as there are only two respondents in this category.

	Number of Responses			Statistic's		
	Valid	%	Missing	Mean	Median	Mode
Analogy	20	45.5	24	4.15	4	5
Expert judgement	27	61.4	17	4.74	5	5
Work breakdown	35	79.5	9	5.29	6	6
Algorithmic model	6	13.6	38	4.50	5	6
Other	2	4.5	42	5.50	5.50	5*

* Multiple modes exist. The smallest value is shown.

Table 10-15 Cost Statistics - Accuracy

Excluding "other" work breakdown is clearly given by 79.5% of respondents as the most accurate technique with a mean of 5.29 and a median and mode of six. Only 4.5% of the respondents to this question indicated work breakdown was less than accurate. The two people who use other techniques have indicated their accuracy is high. Expert judgement with a mean of 4.74 and a median and mode of five was regarded by 61.4% of respondents to more accurate than analogy whose mean was 4.15 and median and mode of four and six respectively. Figure 10-55 once again indicates a clear split of opinion relating to algorithmic techniques with fifty percent of the respondents who answered this question indicating the techniques were very accurate and the other fifty percent indicating a degree of inaccuracy.

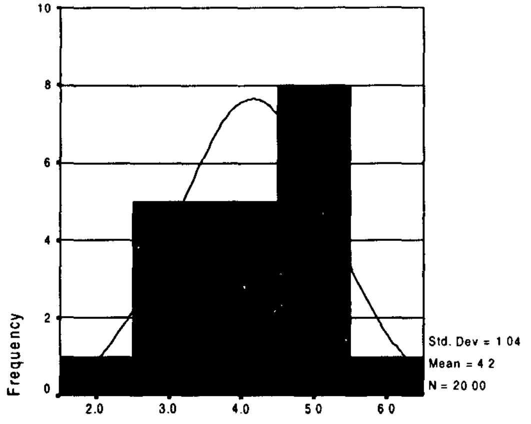

Figure 10-52 Cost - Analogy Accuracy

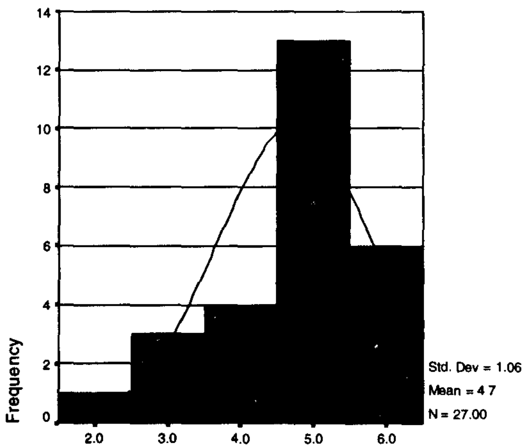

Figure 10-53 Cost - Expert Judgement Accura .y

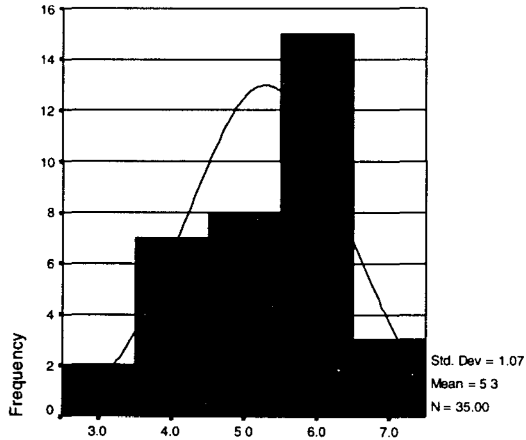

Figure 10-54 Cost - Work breakdown Accuracy

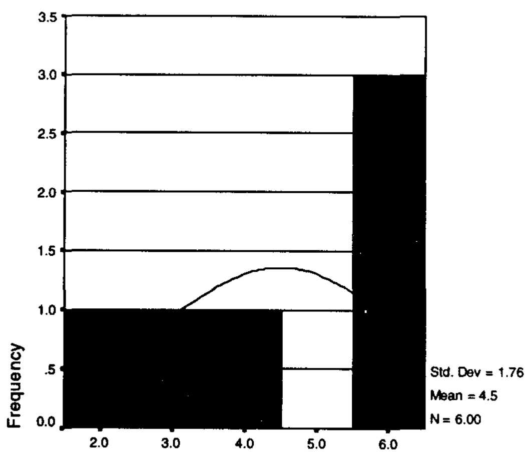

Figure 10-55 Cost - Algorithmic model Accuracy

10.7 Conclusion

Table 10-16 and Figures 10-56 and 10-57 view the data for frequency of use across all the estimating dimensions of size, duration, effort and cost by technique. It can be clearly seen in Figure 10-56 the greatest percentage of respondents use work breakdown in all dimensions. Expert judgement and analogy have the second and third highest use respectively in percentage terms. Algorithmic techniques and other are the least used. It can also be seen in Figure 10-57 that work breakdown is closest to seven on the scale. Seven represented "always" as the frequency of use. This figure also shows the high degree of consistency in the frequency of use of the techniques. It can be observed that effort is the least estimating dimension calculated across all the techniques except cost with analogy where the difference is slight. This raises interesting questions as to how duration is estimated if the effort is unknown or is the duration considered by a significant percentage of practitioners to be a function of the size and/or cost.

	Size		Duration		Effort		Cost	
	%	Mean	%	Mean	%	Mean	%	Mean
Analogy	72.7	4.81	61.4	5.19	50	4.86	47.7	4.71
Expert Judgement	81.8	5.64	75	5.55	56.8	5.28	59.1	5.23
Work breakdown	84.1	5.78	91.1	5.63	65.9	5.69	79.5	5.71
Algorithmic	29.5	3.92	20.5	4.11	15.9	3.86	20.5	4.33
Other	11.4	3.8	11.4	4.2	6.8	3.33	6.8	3.33

Table 10-16 Frequency of use - consolidated

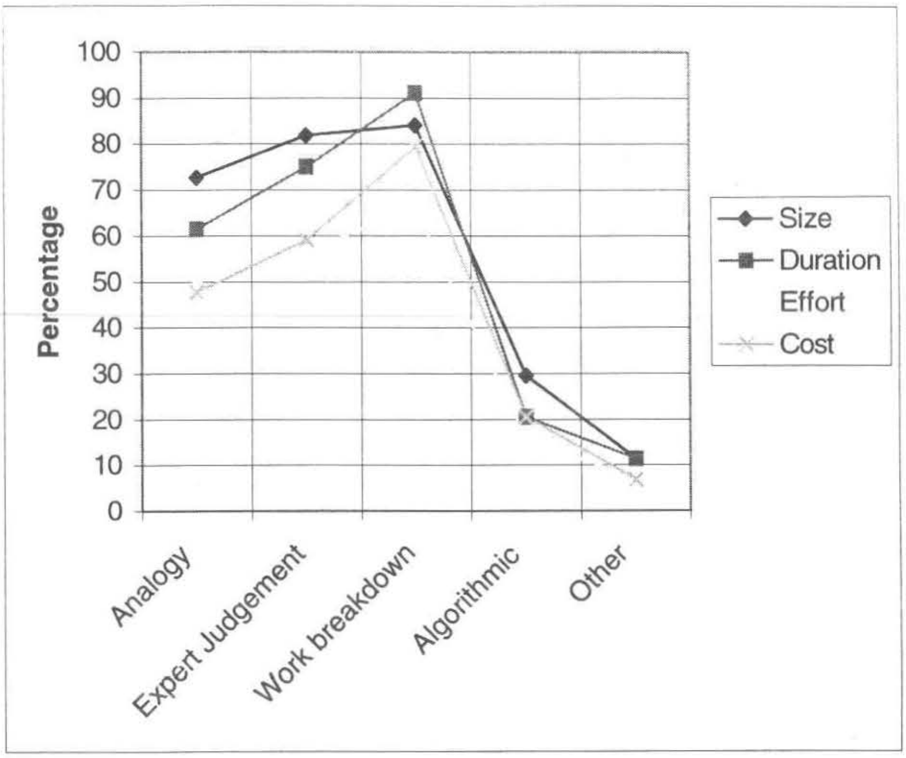

Figure 10-56 Frequency of use percentage by dimension

The percentage of respondents indicating their use of the techniques to measure size shows an excellent correlation with Table 10-1. As the data is derived from separate questions it shows a high degree of consistency in the answers given as they relate to size.

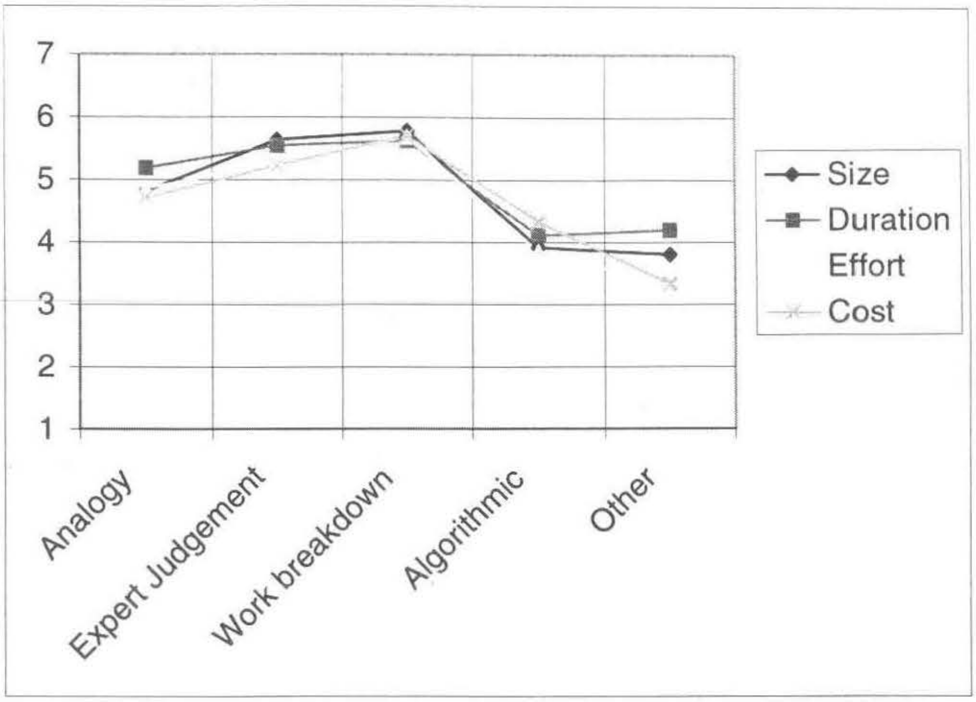

Figure 10-57 Frequency of use means by dimensions

Table 10-17 and Figures 10-58 and 10-59 view the data for ease of use across all the estimating dimensions. Again, it can be seen the greatest percentage of responses was for work breakdown followed by expert judgement, analogy, algorithmic and other. The easiest technique to use, and this is consistent across all dimensions was expert judgement followed closely by the other techniques. Algorithmic and other techniques had the greatest diversity of opinion in the ease of across the dimensions.

	Size		Duration		Effort		Cost	
	%	Mean	%	Mean	%	Mean	%	Mean
Analogy	65.9	4.62	56.8	4.76	47.7	4.67	45.5	4.60
Expert Judgement	77.3	5.15	70.5	5.26	56.8	5.16	61.4	5.04
Work breakdown	77.3	4.82	84.1	4.86	63.6	4.96	77.3	5.12
Algorithmic	25.0	4.27	15.9	4.71	13.6	4.17	13.6	5.00
Other	11.4	3.8	6.8	4.33	4.5	4.50	4.5	4.50

Table 10-17 Ease of use - Consolidated

It should be noted that as discussed early in the chapter the data relating to algorithmic techniques appears to indicate two diverse opinions with one group towards the high end of the scale and the other towards the low.

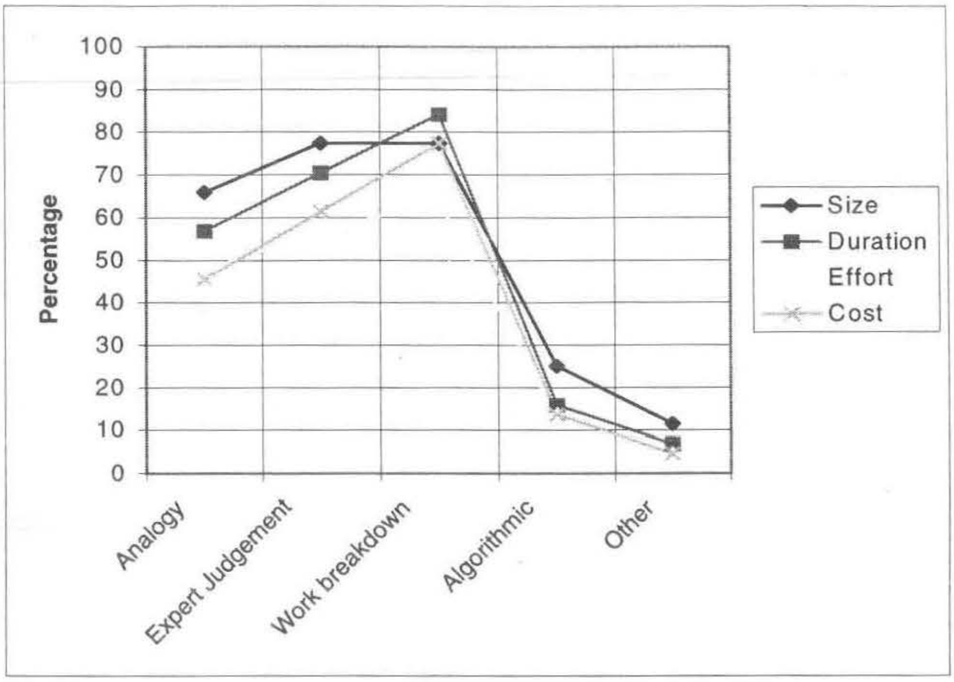

Figure 10-58 Ease of use percentage by dimension

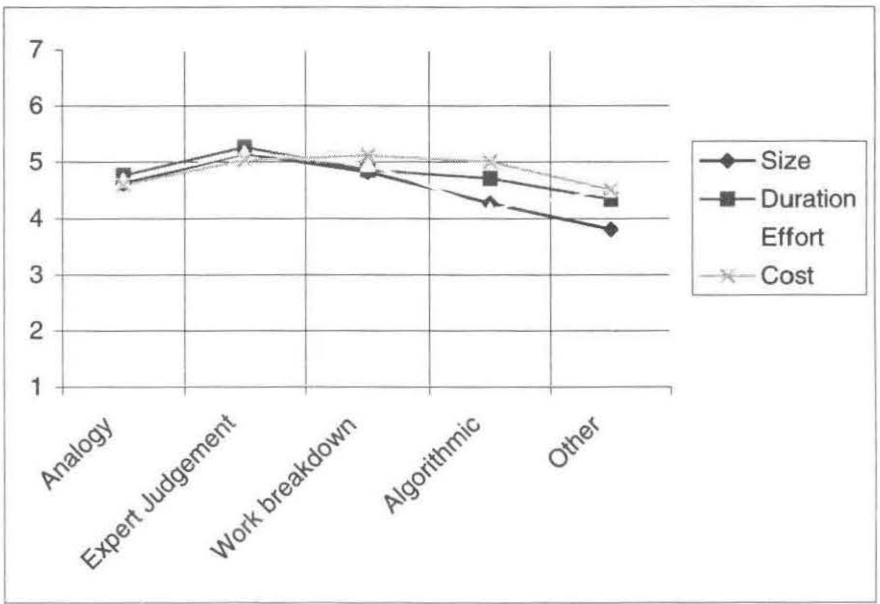

Figure 10-59 Ease of use Means by dimension

Table 10-18 and Figures 10-60 and 10-61 view the data for accuracy across all the estimating dimensions. Again, it can be seen the greatest percentage of responses was for work breakdown followed by expert judgement, analogy, algorithmic and other.

	Size		Duration		Effort		Cost	
	%	Mean	%	Mean	%	Mean	%	Mean
Analogy	68.1	4.27	59.1	4.42	47.7	4.14	45.5	4.15
Expert Judgement	81.1	4.6	72.7	4.66	56.8	4.68	61.4	4.74
Work breakdown	81.1	5.08	88.6	4.95	65.9	5.00	79.5	5.29
Algorithmic	25	4.36	18.2	4.63	13.6	5.17	13.6	4.50
Other	9.1	5	9.1	5	4.5	5.50	4.5	5.50

Table 10-18 Accuracy - Consolidated

Ignoring the "other" category work breakdown is more accurate than the other techniques with the exception of effort estimating where algorithmic techniques are rated higher.

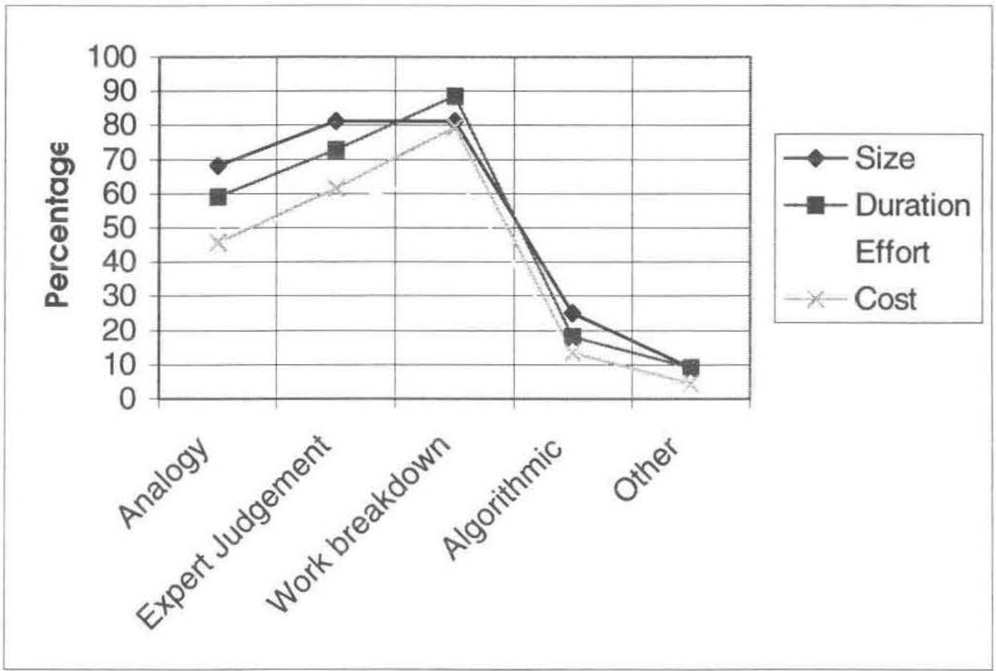

Figure 10-60 Accuracy - Percentage by dimension

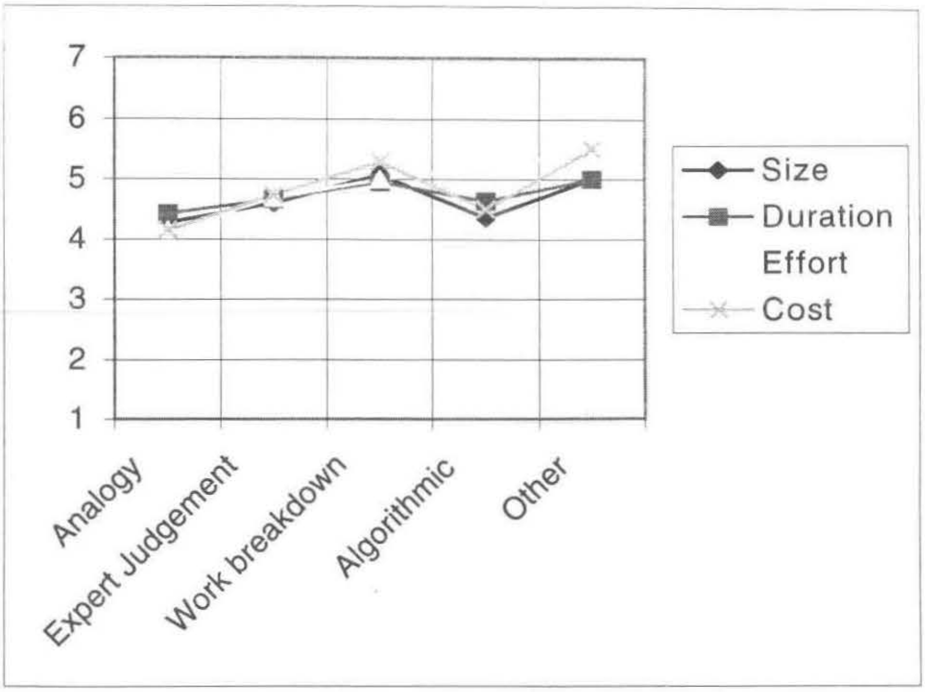

Figure 10-61 Accuracy - Means by dimensions

The data collected shows a high degree of consistency and shows no evidence or trends to indicate the respondents were "just ticking a box".

Chapter 11 Estimating Processes: WA Current Practice

The survey questionnaire also examined the processes that were used in estimating. The formality of the processes was determined by how well they were defined and followed. A formal process would need to be documented in order to be standardised and repeatable. Knowledge regarding the overall formality of the an organisation's processes was determined by the usage of quality management systems.

11.1 Organisational Processes

Table 11-1 and Figure 11-1 show that over one third of the organisations surveyed do not have any form of a Quality Management System (QMS) and about twenty percent have an uncertified QMS. Thirty-four point one percent either have ISO9001 or AS3563.1 and ISO9001 which are the standards applicable for software development. These standards incorporate the design practices whereby ISO 9002 does not. ISO9002 is not suitable for software development.

	QMS Status	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No QMS	15	34.1	34.9	34.9
	Uncertified QMS	9	20.5	20.9	55.8
	ISO9001	8	18.2	18.6	74.4
	ISO9002	3	6.8	7.0	81.4
	AS3563.1 & ISO 9001	7	15.9	16.3	97.7
	Other	1	2.3	2.3	100.0
	Total	43	97.7	100.0	
System Missing		1	2.3		
Total		44	100.0		

Table 11-1 Organisational QMS's

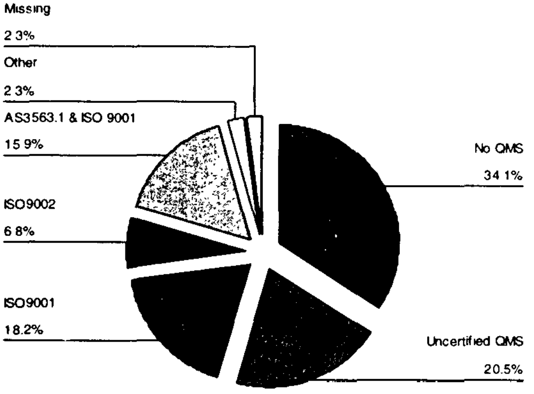

Figure 11-1 Organisational QMS's

Asking whether the organisation had no procedure, an informal procedure or a documented procedure assessed the formality of the procedures for estimating. A truly formal procedure would need to be documented as part of the work practices.

	Estimating Procedure	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No procedure	3	6.8	7.0	7.0
	Informal procedure	30	68.2	69.8	76.7
	Documented procedure	10	22.7	23.3	100.0
Total		43	97.7	100.0	
Missing		1	2.3		
Total		44	100.0		

Table 11-2 Formality of Estimating Procedure

As can be seen from Table 11-2 and the graph in Figure 11-2 the bulk of respondents, 68.2%, have indicated they have an informal procedure. Only ten of the respondents, 22.7%, claim to have a documented procedure. It was expected the procedure formality would correlate well with Table 7-17 where information was sought as to whether the respondent's practices were the same or different to their colleagues. Whilst the percentages are about the same (22.7% and 20.5%) there is no correlation (Spearman's correlation coefficient is 0.176) between having the same practices as a colleague and the formality of the procedure used in an organisation. No significant correlations were found between having the same practices and the formality of the quality management system or development methodology.

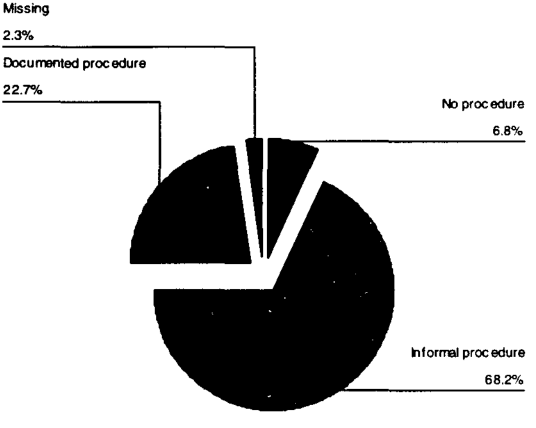

Figure 11-2 Formality of Estimating Procedure

11.2 Information Technology Group Processes

Table 11-3 and Figure 11-3 show that over one third of the organisation's Software House Office or Information Technology group surveyed do not have any form of a Quality Management System (QMS) and 27.3% have an uncertified QMS. A formal QMS certified to ISO9001 and/or AS3563.1 is in 29.5% of the respondent's Software House Office or Information Technology groups.

	QMS Status	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No QMS	16	36.4	36.4	36.4
	Uncertified QMS	12	27.3	27.3	63.6
	Within scope of the org	2	4.5	4.5	68.2
	ISO9001	2	4.5	4.5	72.7
	AS3563.1 & ISO9001	11	25.0	25.0	97.7
	Other	1	2.3	2.3	100.0
Total		44	100.0	100.0	

Table 11-3 Software House or IT Group QMS Status

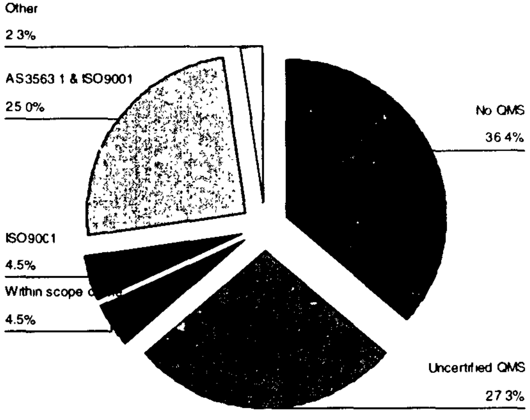

Figure 11-3 Software House or IT Group QMS Status

The formality of the overall development processes was obtained by asking questions relating to methodology usage. This methodology usage was categorised by both new projects and maintenance projects. Table 11-4 and Table 11-5 detail the responses and as can be seen 20.9% do not have a formal method for new developments. For maintenance projects 34.9% do not possess a formal method.

	Development Method Status	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Informal	9	20.5	20.9	20.9
	Formal in-house	27	61.4	62.8	83.7
	Commercial	7	15.9	16.3	100.0
Total		43	97.7	100.0	
Missing		1	2.3		
Total		44	100.0		

Table 11-4 Formality of New Project Methodology

	Development Method Status	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Informal	15	34.1	34.9	34.9
	Formal in-house	23	52.3	53.5	88.4
	Commercial	5	11.4	11.6	100.0
Total		43	97.7	100.0	
Missing		1	2.3		
Total		44	100.0		

Table 11-5 Formality of Maintenance Project Methodology

The lifecycle models used were also of interest to see if they were conducive to the provision of an estimate at the project's beginning. That is, it was surmised that a project that was evolutionary in nature was inherently more difficult to estimate at the beginning. For instance the Spiral Model explicitly defines a process, in each revolution, to form the estimates for the next revolution. See Boehm (1989) for a detailed explanation of the Spiral Model.

However, as can be seen from Table 11-6 the majority of respondents tend to use, for new projects, either a Prototyping approach (77.3%) or the Classic approach (63.6%).

Lifecycle Model	New Projects		Maintenance Projects	
	Frequency	Percent	Frequency	Percent
Waterfall or Classical	28	63.6	25	56.8
Prototyping	34	77.3	18	40.9
Spiral	3	6.8	0	0
Evolutionary	2	4.5	6	13.6
RAD	13	29.5	7	15.9
Object Oriented	8	18.2	3	6.8
Other	0	0	0	0

Table 11-6 Lifecycle Models used

For maintenance projects the Classical approach (56.8%) was preferred over the Prototyping approach (40.9%). RAD is also used a significant amount.

The phase where the initial estimate is required was also sought in order to confirm the hypothesis that estimates are required at project commencement. This is when there are a large number of unknown factors about the project and hence estimating cannot be exact. Table 11-7 clearly demonstrates that this hypothesis to be consistent with the data gathered. That is 81% of respondents are required to produce an estimate before the requirements and designs have been specified.

	Phase	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Quotation	10	22.7	23.8	23.8
	Feasibility study	24	54.5	57.1	81.0
	Requirements analysis	6	13.6	14.3	95.2
	Design	2	4.5	4.8	100.0
Total		42	95.5	100.0	
Missing		2	4.5		
Total		44	100.0		

Table 11-7 Phase of Initial Estimates

11.3 Specific Estimating Processes

A series of questions were posed that defined good practice to ascertain the degree the respondents had implemented estimating processes.

The use of multiple techniques on a project is detailed in Table 11-8 and Figure 11-4. These indicate that multiple techniques are sometimes used on projects although 30.2% were towards the never end of the scale. Only 4.5% of the respondents indicated they always used multiple techniques. This data is supported by that presented in chapter 10 where responses are analysed relating to the different practices used in estimating the different dimensions.

	Rating	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	6	13.6	14.0	14.0
	2	6	13.6	14.0	27.9
	3	1	2.3	2.3	30.2
	4	17	38.6	39.5	69.8
	5	7	15.9	16.3	86.0
	6	4	9.1	9.3	95.3
	7	2	4.5	4.7	100.0
Total		43	97.7	100.0	
Missing		1	2.3		
Total		44	100.0		

Table 11-8 Use of Multiple Techniques on a Project

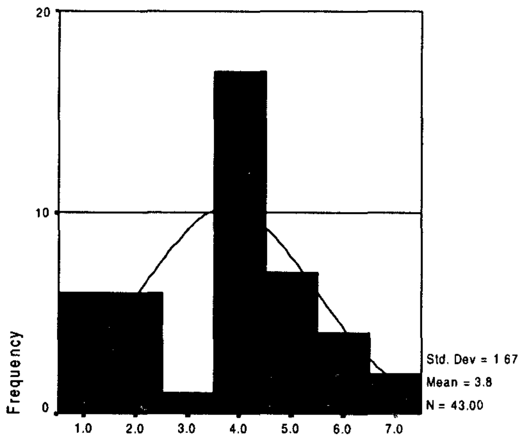

Figure 11-4 Use of Multiple Techniques on a Project

Table 11-9 and Figure 11-5 show the respondents tend to keep records of their estimates although only 30.2 % advised they always kept records.

	Rating	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	2	4.5	4.7	4.7
	2	1	2.3	2.3	7.0
	3	2	4.5	4.7	11.6
	4	6	13.6	14.0	25.6
	5	7	15.9	16.3	41.9
	6	12	27.3	27.9	69.8
	7	13	29.5	30.2	100.0
Total		43	97.7	100 0	
Missing		1	2.3		
Total		44	100.0		

Table 11-9 Frequency of Record Keeping

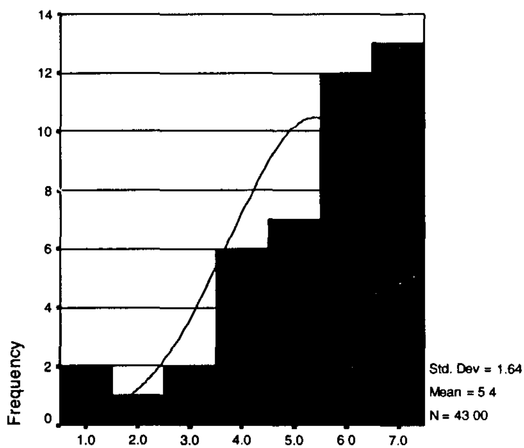

Figure 11-5 Frequency of Record Keeping

Respondents tended not to indicate a probability of achievement for their estimates and the data is shown in Table 11-10 and Figure 11-6. The table shows 34.9% of respondents never give a probability of achievement and 81.4% indicated sometimes or less on the scale. Only one respondent representing 2.3% advised they always gave a probability.

	Rating	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	15	34.1	34.9	34.9
	2	9	20.5	20.9	55.8
	3	4	9.1	9.3	65.1
	4	7	15.9	16.3	81.4
	5	4	9.1	9.3	90.7
	6	3	6.8	7.0	97.7
	7	1	2.3	2.3	100.0
Total		43	97.7	100.0	
Missing		1	2.3		
Total		44	100.0		

Table 11-10 Indication of Probability of Achievement

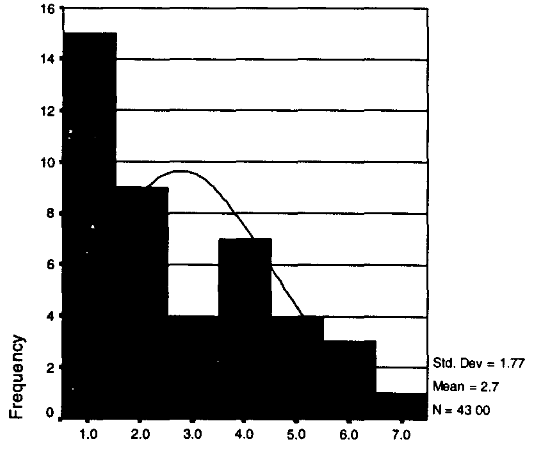

Figure 11-6 Indication of Probability of Achievement

Respondents tended not to use a range for their estimates and the data is shown in Table 11-11 and Figure 11-7. The table shows 16.3% of respondents never give a range and 74.4% indicated sometimes or less on the scale. Only one respondent representing 2.3% advised they always gave a range.

	Rating	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	7	15.9	16.3	16.3
	2	7	15.9	16.3	32.6
	3	7	15.9	16.3	48.8
	4	11	25.0	25.6	74.4
	5	6	13.6	14.0	88.4
	6	4	9.1	9.3	97.7
	7	1	2.3	2.3	100.0
Total		43	97.7	100.0	
Missing		1	2.3		
Total		44	100.0		

Table 11-11 Use of Estimating Ranges

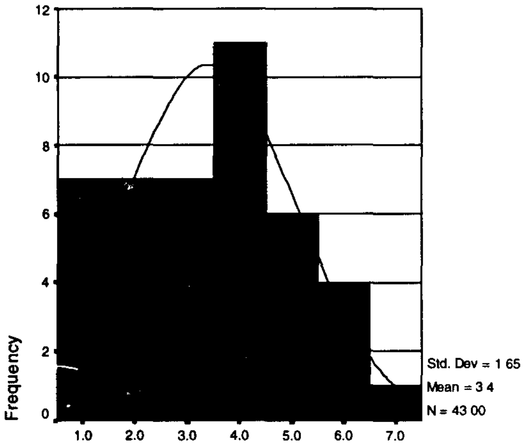

Figure 11-7 Use of Estimating Ranges

Respondents tended to have peers review their estimates and the relevant data is shown in Table 11-12 and Figure 11-8. The table shows only 6.8% of respondents never undertook peer reviews although approximately half (53.5%) indicated sometimes or less on the scale. Four respondents representing 9.1% advised they always had their estimates peer reviewed.

	Rating	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	3	6.8	7.0	7.0
	2	1	2.3	2.3	9.3
	3	6	13.6	14.0	23.3
	4	13	29.5	30.2	53.5
	5	6	13.6	14.0	67.4
	6	10	22.7	23.3	90.7
	7	4	9.1	9.3	100.0
Total		43	97.7	100.0	
Missing		1	2.3		
Total		44	100.0		

Table 11-12 Peer Reviewing of Estimates

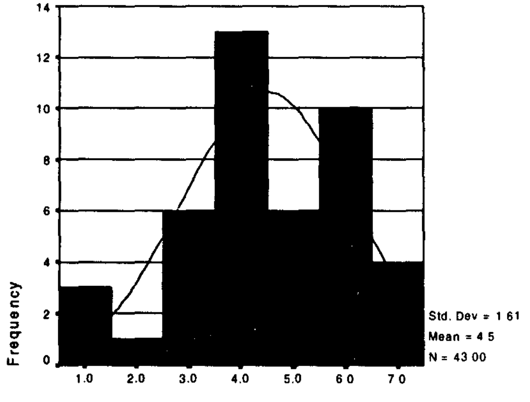

Figure 11-8 Peer Reviewing of Estimates

It would appear from the data in Table 11-13 and Figure 11-9 that it is accepted practice to review estimates. Only one respondent representing 2.3% indicated they never reviewed their estimates. Approximately a third (31.8%) have estimates constantly under review and 47.7% reconsider their estimates as part of their project reviews. A minority 15.9% indicated they reviewed the estimates only as the specifications changed. This could perhaps be construed as constantly under review if they are using a Prototyping approach as 77.3% of respondents advised they did for new projects.

	Frequency	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	never	1	2.3	2.3	2.3
	Only as specifications change	7	15.9	16.3	18.6
	At project review points	21	47.7	48.8	67.4
	Constantly under review	14	31.8	32.6	100.0
Total		43	97.7	100.0	
Missing		1	2.3		
Total		44	100.0		

Table 11-13 Frequency of Estimate Revision

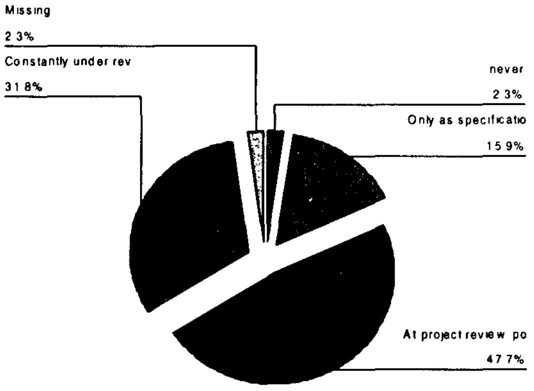

Figure 11-9 Frequency of Estimate Revision

Respondents tended to compare estimates with actual results as can be seen in Table 11-14 and Figure 11-10. There was no response indicating it was never done and the majority, 61.5% indicated they did comparisons more frequently that sometimes. The mean of five and modes of five, six and seven also add weight to the assertion that respondents tend to compare actuals with estimates.

	Frequency Rating	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	1	2.3	2.3	2.3
	3	7	15.9	16.3	18.6
	4	8	18.2	18.6	37.2
	5	9	20.5	20.9	58.1
	6	9	20.5	20.9	79.1
	7	9	20.5	20.9	100.0
Total		43	97.7	100.0	
Missing		1	2.3		
Total		44	100.0		

Table 11-14 Comparison of Estimates with Actuals

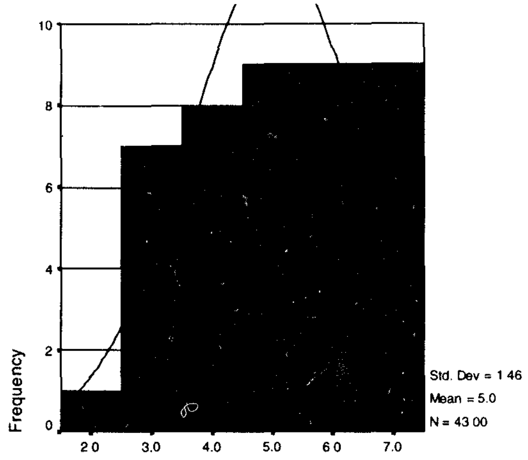

Figure 11-10 Comparison of Estimates with Actuals

Formal feedback mechanisms tend not to be used with 53.5% indicating they were used less than sometimes. Table 11-15 and Figure 11-11 show the data. Only 4.7% advised they always used a formal feedback mechanism.

	Frequency Rating	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	8	18.2	18.6	18.6
	2	10	22.7	23.3	41.9
	3	5	11.4	11.6	53.5
	4	10	22.7	23.3	76.7
	5	6	13.6	14.0	90.7
	6	2	4.5	4.7	95.3
	7	2	4.5	4.7	100.0
Total		43	97.7	100.0	
Missing		1	2.3		
Total		44	100.0		

Table 11-15 Use of Formal Feedback

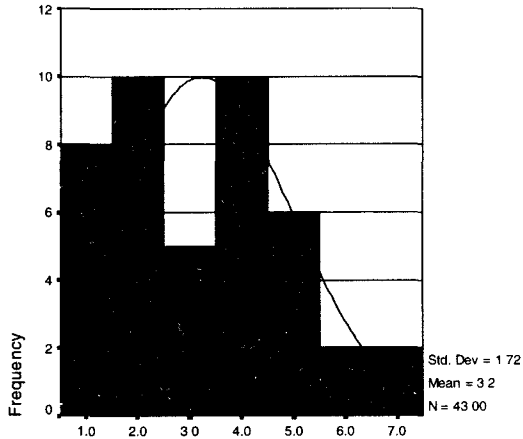

Figure 11-11 Use of Formal Feedback

11.4 Conclusions.

The formality of the processes whether they be

- quality management system
- methodology
- defined estimating procedure
- use of multiple techniques
- record keeping
- indication of probability of achievement
- indication of a range
- conducting of peer reviews

appear to have little influence on the accuracy or degree of confidence the respondents expressed in their estimates.

Correlations are shown in Table 11-16 using Pearson's correlation coefficient. Only those correlations that are significant at the .05 level (two-tailed) are shown.

	Use of estimating ranges	Peer reviewing of estimates	Frequency estimate/actual comparison	Accuracy of the organisation's estimates
Use of multiple estimating techniques				.387
Estimate records kept			.333	
Probability of achievement indication	.474			
Use of ranges		.455		
Peer reviewing of estimates			.506	
Frequency estimate/actual comparison				
Degree of satisfaction with estimating process				.737

Table 11-16 Significant Process correlations

Chapter 12 Conclusion

The research has provided data and information that answered the questions posed in chapter 1 and also uncovered a number of interesting areas, some of which warrant further exploration.

The survey conducted, as part of this research, resulted in demographic data comparable to other surveys such as Wydenbach and Paynter (1995) and Hihn and Habib-agahi (1991) as described in chapter 5. For instance, Hihn and Habib-agahi (1991) found in their survey that the respondents had a mean of 14.9 years of experience and a standard deviation of 7.6 years. The data from the WA survey shows a mean of 15.8 years of experience and a standard deviation of 5.9 years - see Figure 7-11. Therefore one can conclude the respondents to the WA survey were slightly more experienced than the respondents to Hihn and Habib-agahi's survey. They also conduct more estimates per annum as Figure 7-19 shows a mean of 5.7 estimates for the last twelve months. This is an estimate every two months versus Hihn and Habib-agahi's (1991) respondents who completed an estimate every eight months. Data and results in the other surveys conducted and reported in chapter 5 are similar to the results of this survey in comparable areas.

The WA survey also showed that respondents:

- had used a wide variety of techniques in their careers
- had worked for several organisations
- came from a wide variety of organisational types and sizes.

It was also shown, like the other surveys, only a small percentage of practitioners use algorithmic estimating methods and most use a form of work breakdown, expert judgement and analogy. Table 5-1 shows, from other surveys conducted that the use of algorithmic techniques ranges from 14% to 30%. These figures correlate well with the WA survey data that shows a range from 15.9% to 29.5% (see Table 10-16) depending on what is being estimated - size, duration, effort or cost.

Therefore, it can be said that the software estimating practices in WA are similar to counterparts in other parts of the world. It can also be said that the opinions and information given by the respondents to the WA survey has been given by experienced practitioners. Furthermore the respondents advised their practices were similar to their colleagues as shown in Figure 7-20. For these reasons conclusions drawn from this survey can be said to apply to the population as a whole.

As described in Table 10-1 and Figure 10-2 the survey respondents overall tend to mostly use a work breakdown (84.1%) in the formation of their estimates. This is closely followed by the use of expert judgement (72.8%) and analogy (70.5%) with algorithmic techniques being used by 24.6%. The frequency of use of the work breakdown technique, as shown in Table 10-4, is also the highest rated of any technique. The survey respondents advised they are also using expert judgement and analogy however these techniques do not appear to used in a formal sense with a documented and validated approach. It can also be said that although 15.9% (Figure 10-1) advised they had discarded any techniques, 50% advised they had used lines of code techniques and 52.3% indicated they had used function point analysis at some stage in their career - (see Tables 7-12 and 7-13). However as 24.6% advised they still used algorithmic techniques one must conclude that although the respondents are aware of the techniques and have not completely discarded them they rarely, if ever, use them. This assertion is supported by the data in Table 10-4 and Figure 10-7 which indicate the frequency of use algorithmic techniques is lower that that of other techniques. An effort was made to determine if different techniques were used when estimating the different project dimensions of size, effort, duration and cost The data shows different techniques are not used. The data raises the question of how the size of a project is measured, as the respondents are not using, in the main, lines of code or functional measures.

The elements that are mostly always considered by the respondents in forming an estimate match the classic lifecycle phases. Other elements such as environmental changes, hardware and systems software are considered to a lesser extent. The level of confidence that the respondents have in their estimates is highest for the initial project phases and activities such as requirements analysis and project management. The confidence of the estimates produced is lowest for those elements that are considered less often. See chapter 8 and especially Table 8-13 for the supporting data.

The high level elements proposed by Hope (1996) appear to have validity in that few respondents indicated other elements that they estimated.

The parameters that cause respondents to modify their nominal estimates were also determined. The parameters that have the most impact in terms of the percentage who always consider the parameters and the degree of impact are people skills, knowledge of the application domain, complexity of the problem, project risk and schedule constraints. See Table 9-19 for a summary of the data An interesting observation is that one of the parameters that had the least effect, in terms of percentage who considered and degree of impact, was the size of the database. As can be seen from the discussion in chapter 2 database size is the principle driver of function point analysis. The respondents to the WA survey appear not to rate this aspect highly as an input to their estimates.

The practitioners' opinions as to the strengths and weaknesses of the various techniques were not obtained in detail. However, it would appear, as shown in Table 10-2 and Figure 10-3 that the principle reasons for usage of work breakdown, expert judgement and analogy are that they are appropriate for the projects, easy to use and found to be accurate. The availability of training was not a significant reason for technique usage.

Table 5-7 shows the requirement for the formation of estimates at project initiation to be the case in the USA and the WA data in Table 11-7 supports this view with 54.5% indicating that an estimate is first required at the feasibility study stage of a project. It should be noted these estimates are required at project initiation when there is insufficient data to apply either function point analysis or COCOMO. Table 7-23 adds further weight to this as it shows estimates are required before requirements ie to obtain project approval.

It can be said that good estimating practice is defined by having a documented process, use of multiple techniques, giving ranges and probabilities of achievement, keeping records and peer reviewing the estimates (Boehm 1981, Pressman 1997). It would appear that good practice is not the norm in the organisations surveyed. For instance only 22.7% had a documented estimating procedure and they tended not to give a probability of achievement or range for the estimates or use a formal feed back mechanism - see chapter 11 for details. However, they did tend to keep records, compare estimates with actuals and have the estimates peer reviewed. It would appear to be acknowledged by the respondents that estimates are going to change and they need constant reviewing. Table 11-13 shows this with only one respondent representing 2.3% indicating they never reviewed their estimates.

It would appear that the most popular, at least in the literature, estimating techniques of COCOMO and function point analysis are not widely used in practice. More so these techniques are flawed in their foundations. COCOMO with its reliance on lines of code and function point analysis for several reasons - principally its mathematical manipulation of the scale types it uses. It can also be said there has been little validation of their predicative capabilities and one such study by Kemerer (1987) p416 - 429 showed COCOMO estimates to out by in excess of 600%.

Existing algorithmic techniques are only suited to the software component of a project. They do not cater for all the elements of software intensive projects. The survey indicates that other elements must be considered, such as the development and operational hardware, in the development of the overall system.

Some of the observations that can be made from the survey data are that:

- The assertions by Boehm (COCOMO 2 1995) regarding the development paradigm changing with a greater use of packages is supported by Table 7-1's data.

- Table 7-19 and associated discussion shows maintenance work (40%) versus new development (60%) work is contrary to accepted wisdom. The literature such as Takang and Grubb (1996) typically report a ratio of 60% maintenance and 40% new development. This may also support the view that organisations are moving towards packaged solutions and hence there is less need for maintenance work.
- There is a need for a technique that can be applied early in the lifecycle as the survey data indicates the feasibility study stage is when an estimate is mostly required.
- Estimating techniques are required that suit both a Classical approach to software development and a Prototyping approach where the product evolves. Table 11-6 shows data that supports this. There is of course the difficulty in predicting outcomes if the product under development is being defined and redefined.
- Table 7-20 shows the importance of estimating to an organisation however it would appear that the process is not well formalised. Table 11-2 and Figure 11-2 demonstrate the lack of a documented process.
- The data in Tables 7-21 and 7-22 show the respondents are slightly more than satisfied with the estimating process within the organisation and view the estimates as being more than somewhat accurate. This leads to the conclusion that the techniques that are prevalent work breakdown, analogy and expert judgement are providing the results they want.

The following are some recommendations resulting from this work:

- The formalising of an estimating process may be beneficial in order to achieve a disciplined, repeatable approach that collects actual data that can be used for future estimates. Nine of the respondents to the survey suggested having a formalised process or method would improve estimating within their organisation.
- Having a historical database record of estimates and actuals was one of the clear suggestions coming from twenty of the respondents to the WA survey. Also the respondents to Park, Goethert & Webb's (1994) survey (see chapter 5) suggested this was one of the ways estimating would be improved. Therefore it can be seen that their findings concur with the WA survey.
- Giving a probability of achievement for an estimate would assist in managing the perception that users have of the estimate being a firm quote and not an estimate. This is not done as evidenced by Table 11-10. This may be difficult to achieve in practice with fixed price contacts and organisations' budgeting and cash flow needs demanding firm quotations.

In conclusion one must agree with Abdel-Hamid (1993) p20 that the difficult problem of software estimation remains unresolved. D'Marco (1982) p29 suggested the development of an estimator's handbook to be refined over time. As the survey data indicates practitioners are developing detailed work breakdowns and this is the preferred technique, a formal detailed workbook method for estimating software projects should be developed and refined over a period of time.

References

Abdel-Hamid, T. K. (1993). Adapting, Correcting, and Perfecting Software Estimates: A Maintenance Metaphor. Vol 26, No 3 Computer, IEEE Computer Society.

Abran, A., Robillard, P.N. (1996). Function Points Analysis: An Empirical Study of Its Measurement Process. Vol 22, No 12 IEEE Transactions on Software Engineering, IEEE Computer Society.

Albrecht, A.J. (1979). Measuring Application Development Productivity. Proceedings - Joint Share/Guide IBM Application Development Symposium pp 83-92.

Australian Bureau of Statistics (1993). 1292.0.15.002 Australian and New Zealand Standard Industrial Classification (ANZSIC) - Titles and Codes.

Banker,R., Kemerer, C. (1989). Scale Economies in New Software Development. Vol 15, No 10 IEEE Transactions on Software Engineering, IEEE Computer Society.

Basili, V.R., Rombach, D. (1988). The TAME Project. Towards Improvement-oriented Software Environments. Vol 14, No 6 IEEE Transactions on Software Engineering, IEEE Computer Society.

Boehm, B.W. (1981). Software Engineering Economics. New York: Prentice Hall.

Boehm, B.W. (1984). Software Engineering Economics. Vol 10, No 1 IEEE Transactions on Software Engineering, IEEE Computer Society.

Boehm, B.W. (1987). Ada COCOMO: TRW IOC version.. Third COCOMO User's Group Meeting.

Boehm, B.W. (1992), Risk Control, American Programmer Vol 5 No 7, pp 2-9, New York, NY.

Boehm, B.W., Clark, B., Horowitz, E., Westland, C. (1996). The COCOMO 2.0 Software Cost Estimation Model: A Status Report. American Programmer Vol 9 No 7, pp 2-17, New York, NY.

Borg, W. R., Gall, M. D. (1979). Educational Research. - Third Edition. New York: Longman, Inc.

Briand, L., El Amam, K., Bomarius, F (1997). A Hybrid Method for Software Cost Estimation and Risk Assessment. ISERN-97-24. International Software Engineering Research Network.

Brooks, F.P. (1975). The Mythical Man Month. Addison-Wesley

Charette, R.N. (1989). Software Engineering Risk Analysis and Management. McGraw-Hill Book Company, New York, NY.

COCOMO (1995). COCOMO 2.0 Model User's Manual - Version 1.1. University of Southern California.

Daskalantonakis, M. K. (1992). A Practical View of Software Measurement and Implementation Experiences within Motorola. Vol 18, No 11 IEEE Transactions on Software Engineering, IEEE Computer Society.

DeMarco, T. Lister, T. (1987). Peopleware: Productive Projects and Teams. Dorset House.

Dreger, J. (1989). Function Point Analysis. Englewood Cliffs, NJ: Prentice Hall.

Fenton, N. (1994). Software Measurement: A Necessary Scientific Basis. Vol 20, No 3 IEEE Transactions on Software Engineering, IEEE Computer Society.

Fenton, N. E. (1991). Software Metrics - A Rigorous Approach. London: Chapman & Hall.

Garmus, D., Herron, D., (1996). Measuring the Software Process: A Practical Guide to Functional Measurements Englewood Cliffs, NJ: Prentice Hall.

Gilb, T. (1988). Principles of Software Engineering Management. Wokingham: Addison-Wesley.

Hihn, J., Habib-agahi, H. (1991). Cost Estimation of Software Intensive Projects: A Survey of Current Practices. 13th International Conference on Software Engineering, IEEE Computer Society.

Hope, S. (1996). Towards a Model for Software Project Estimating. Honours Thesis. Edith Cowan University.

Hope, S. (1997). Software Project Estimating. APT Methodology Release 6.0. Execom.

Humphrey, W.S. (1989). Managing the Software Process. Reading, Massachusetts: Addison-Wesley.

Humphrey, W.S. (1995). A Discipline for Software Engineering. Reading, Massachusetts: Addison-Wesley.

IFPUG, (1994). Function Point Counting Practices Manual, Release 4.0. International Function Point Users Group, Westerville: Ohio.

ISO/IEC 12207:1995. Information Technology - Software Lifecycle Processes.

Jackson, M. (1975). Principles of program Design. London, Academic.

Jeffery, D. R., Low, G.C., Barnes, M. (1993). A Comparison of Function Point Counting Techniques. Vol 19, No 5 IEEE Transactions on Software Engineering, IEEE Computer Society.

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

Jorgensen, M. (1995). Experience With the Accuracy of Software Maintenance Task Effort Prediction Models. Vol 21, No 8 IEEE Transactions on Software Engineering, IEEE Computer Society.

Judd, C.M., Smith, E.R., Kidder, L.H. (1991). Research Methods in Social Relations Sixth Edition. Harcourt Brace Jovanovich College Publishers.

Kemerer C.F. (1987). An Empirical Validation of Software Cost Estimation Models. Vol 30 No 5, Communications of the ACM.

Kemerer, C. F., Porter, B. S. (1992). Improving the Reliability of Function Point Measurement: An Empirical Study. Vol 18, No 10 IEEE Transactions on Software Engineering, IEEE Computer Society.

Kitchenham, B. (1992). Empirical Studies of Assumptions that Underlie Software Cost-estimation Models. Vol 34 No 4 Information and Software Technology , Butterworth-Heinemann Ltd.

Kitchenham, B., Kansala, K. (1993). Inter-item Correlations among Function Points. Proceedings of the IEEE Metrics Symposium, IEEE Computer Society.

Kitchenham, B., Pfleeger, S.L., Fenton, N. (1995). Towards a Framework for Software Measurement Validation. Vol 21, No 12 IEEE Transactions on Software Engineering, IEEE Computer Society.

Lederer, A.L., Prasad, J. (1998). A Causal Model for Software Cost Estimating Error. Vol 24, No 2 IEEE Transactions on Software Engineering, IEEE Computer Society.

Leedy, P.D. (1993). Practical Research: planning and design - 5th edition. Englewood Cliffs, NJ. Macmillian Publishing Co

Matson, J. E., Barrett, B. E., Mellichamp, J. M (1994). Software Development Cost Estimation Using Function Points. Vol 20, No 4 IEEE Transactions on Software Engineering, IEEE Computer Society.

McCabe, T. J. (1976). A Complexity Measure. Vol 2, No 4 IEEE Transactions on Software Engineering, IEEE Computer Society.

Mukhopadhyay, T., Kekre, S. (1992). Software Effort Models for Early Estimation of Process Control Applications. Vol 18, No 10 IEEE Transactions on Software Engineering, IEEE Computer Society.

Park, R.E., Goethert, W.B., Webb, J.T. (1995). Software Cost and Schedule Estimating: A process Improvement Initiative. Special Report CMU/SEI-94-SR-3, Carnegie Mellon University.

Parnas, D.L. (1994). Software Aging. Proceedings of the 16th International Conference on Software Engineering - ICSE '94. IEEE Computer Society

Pressman, R.S. (1997). Software Engineering: A Practitioner's Approach 4th Edition. McGraw-Hill, Inc.

Putnam, L.H., Myers, W. (1992). Measures for Excellence: Reliable Software on Time, within Budget. Englewood Cliffs, NJ: Prentice Hall.

Quinn, J.B., Baruch, J.J., Zien, K.A. (1997). Innovation Explosion: Using Intellect and Software to Revolutionise Growth Stategies. New York, The Free Press.

Rosenthal, R., Rosnow, R. L. (1975). The Volunteer Subject. New York: John Wiley

Ross. K.C. (1996). Air University Sampling and Surveying Handbook Internet Edition. United States Government, Department of Defence, the Department of the Air Force.

Rutherford, J. (1995). Software Estimating and Metrics - A Pragmatic Approach. Vol 12, No 2 The GEC Journal of Research, General Electric Company

Shepperd, M., Schofield, C. (1997). Estimating Software Project Effort Using Analogies. Vol 23, No 12 IEEE Transactions on Software Engineering, IEEE Computer Society.

Siegel, S. (1956). Nonparametric Statistics for the Behavioural Sciences. McGraw-Hill, Inc.

St-Pierre, D., Maya, M., Abran, A., Desharnais, J-M., Bourque, P. (1997). Full Function Points: Counting Practices Manual. Technical Report 1997-04. University of Quebec.

Symons, C.R. (1988). Function Point Analysis: Difficulties and Improvements. Vol 14, No 1 IEEE Transactions on Software Engineering, IEEE Computer Society.

Takang, A. A., Grubb, P. A. (1996). Software Maintenance Concepts and Practice. London, International Thomson Computer Press.

Thomsett, R. (1991). Managing Superlarge Projects: A Contingency Approach. Vol 4, No 6 American Programmer, American Programmer Inc.

Thomsett, R. (1993). Third Wave Project Management. Englewood Cliffs, NJ: Prentice Hall.

Verner, J. Tate G. (1992). A Software Size Model. Vol 18, No 4 IEEE Transactions on Software Engineering, IEEE Computer Society.

Weinberg, G.M. (1993). Quality Software Management: Volume 2 First Order Measurement. New York: Dorset House Publishing.

Wydenbach, G., Paynter, J. (1995). Software Project Estimation: A Survey of Practices in New Zealand. Technical report No 97, University of Auckland.

Appendix 1 - Survey Questionnaire

A Survey of Estimating Practices for Software Intensive Projects

Organisation ID ☐

You will receive a copy of the survey report if you complete the details in this box. Note information provided here will be kept confidential and will NOT be used in the report.

Organisation Name: ______________________________
Address: ______________________________
Contact person: ______________________ **Contact Number:** ______________________

General Instructions:
Where indicated please place a cross in the box or boxes of your choice. With scales please circle a number.

Part A: Personal Details

A.1 Please indicate the number of years in the current organisation. ☐

A.2 Indicate the total number of years experience you have in software development. ☐

A.3 Indicate the software project estimating techniques that you have used in **your** career. Please also rate their usefulness by circling a number. (1=useless, 4=useful; 7=very useful).

☐ **Analogy**
(To compare past projects and extrapolate to new projects)
1 2 3 4 5 6 7

☐ **Expert Judgement**
(To obtain expert opinions and apply to new projects)
1 2 3 4 5 6 7

☐ **Work Breakdown**
(To define all the tasks in a project, estimate individually and then aggregate)
1 2 3 4 5 6 7

☐ **Lines of Code techniques**
(To estimate the number of source instructions that will be delivered eg COCOMO)
1 2 3 4 5 6 7

☐ **Function Point Analysis**
(Using Albrecht's techniques or a derivative)
1 2 3 4 5 6 7

☐ **Other** (please specify): ______________
1 2 3 4 5 6 7

A.4 Please rate your software project estimating skills. (1=poor; 4=average; 7=excellent).

1 2 3 4 5 6 7

A.5 Approximately how many total project estimates have you made in the last twelve months? ☐

Part B: Organisational Information.

B.1 Please indicate if the organisation has a quality management system and its certification status.
- ☐ No Quality Management System
- ☐ Uncertified Quality Management system
- ☐ ISO 9001 certified
- ☐ ISO 9002 certified
- ☐ ISO 9003 certified
- ☐ AS 3563.1 certified
- ☐ Other. (please specify):
  ____

B.2 Please insert the organisation's approximate total number of full time employees in Australia. ☐

B.3 Please insert the organisation's approximate total number of part time employees in Australia. ☐

B.4 Please indicate your organisation's industry sector(s)?
- ☐ agriculture
- ☐ mining
- ☐ manufacturing
- ☐ electricity/gas/water
- ☐ construction
- ☐ wholesale trade
- ☐ retail trade
- ☐ accommodation
- ☐ transport/storage
- ☐ communications
- ☐ finance/insurance
- ☐ property and business services (including software & consultancy houses)
- ☐ government/defence
- ☐ education
- ☐ health and community services
- ☐ cultural and recreational services
- ☐ personal and other services

Part C: Software House Office or Information Technology Group Information.

C.1 Please indicate if the office or Information Technology Group has a Quality Management System.

No Quality Management System
Uncertified Quality Management system
Within the scope of the organisation
ISO 9001 certified
AS 3563.1 certified
Other. (please specify):

____

C.2 Please insert the approximate number of people developing and maintaining software (including contractors). ☐

C.3 Please indicate the approximate number of contractors in C.2 above. ☐

C.4 Please indicate, by percentage, the types of software developed by the organisation.

☐ applications for internal use

☐ applications for external clients

☐ commercial packages

☐ other - please specify ______________________

Total 100%

C.5 Please indicate the approximate percentage of software projects undertaken by the organisation within the following categories.

☐ Business

☐ Engineering & Scientific

☐ System

☐ Real time/Embedded

☐ Other (please specify): ___________

Total 100%

Part D: Software Estimation Practices.
NOTE: Some questions have NEW and MAINTENANCE project dimensions.

D.1 Do your estimation practices differ from that of your colleagues? (1=Same; 4=somewhat different; 7=very different)

1 2 3 4 5 6 7

D.2 Do you use different practices for different project categories? (1=Same; 4=somewhat different; 7=very different)

1 2 3 4 5 6 7

D.3 Please state what percentage of development work is

New development ___________
Maintenance ___________
Total **100%**

D.4 Please indicate which software development methodology is used.

New		**Maintenance**
•	Informal	•
•	Formal in-house	•
•	Commercial	•

D.5 Please indicate the software development lifecycle(s) used.

New		**Maintenance**
•	Waterfall or classical	•
•	Prototyping	•
•	Spiral	•
•	Evolutionary	•
•	RAD	•
•	Object Oriented	•
•	Other (please name): ______________________	•

D.6 Please indicate who are **involved** in formulating the **initial** estimates for projects.

New		**Maintenance**
•	Consultants	•
•	IT Management	•
•	Client Management	•
•	Client Users	•
•	Project Managers	•
•	Analysts	•
•	Programmers	•
•	Specialist Estimating Staff	•
•	Sales People	•
•	Other (please specify): ______________________	•

D.7 What lifecycle phases and other elements are typically included in the scope of your **initial nominal** project estimates? Please indicate the elements considered by crossing a box. Please also indicate your normal level of confidence in the estimate for the particular element by circling a number.

Element	**Considered** never	sometimes	always	**Degree of confidence** (1=not confident, 4=somewhat confident, 7=extremely confident)
Problem Definition & Feasibility Study	•	•	•	1 2 3 4 5 6 7
Requirements Analysis	•	•	•	1 2 3 4 5 6 7
Design	•	•	•	1 2 3 4 5 6 7
Development (coding)	•	•	•	1 2 3 4 5 6 7
Implementation	•	•	•	1 2 3 4 5 6 7
Training	•	•	•	1 2 3 4 5 6 7
Project Management & Administration	•	•	•	1 2 3 4 5 6 7
Development Hardware and Software	•	•	•	1 2 3 4 5 6 7
Operational Hardware & Software	•	•	•	1 2 3 4 5 6 7
Environmental Changes	•	•	•	1 2 3 4 5 6 7
Maintenance	•	•	•	1 2 3 4 5 6 7
Other (please list):				
______________________		•	•	1 2 3 4 5 6 7
______________________		•	•	1 2 3 4 5 6 7
______________________		•	•	1 2 3 4 5 6 7
______________________		•	•	1 2 3 4 5 6 7
______________________		•	•	1 2 3 4 5 6 7
______________________		•	•	1 2 3 4 5 6 7
______________________		•	•	1 2 3 4 5 6 7

D.8 Please indicate the parameters which would cause the nominal estimates for projects to be modified. Indicate the parameters considered and the degree of impact they have on the nominal estimate.

Parameter	**Considered** never	sometimes	always	**Degree of impact** (1=not significant; 4=significant, 7= very significant)
People skills	•	•	•	1 2 3 4 5 6 7
Knowledge of the application domain	•	•	•	1 2 3 4 5 6 7
Complexity of the problem	•	•	•	1 2 3 4 5 6 7
Algorithmic complexity	•	•	•	1 2 3 4 5 6 7
Stability of the target platform	•	•	•	1 2 3 4 5 6 7
User support	•	•	•	1 2 3 4 5 6 7
Training	•	•	•	1 2 3 4 5 6 7
Maintainability	•	•	•	1 2 3 4 5 6 7
Performance requirements	•	•	•	1 2 3 4 5 6 7
Usability	•	•	•	1 2 3 4 5 6 7
Data base size	•	•	•	1 2 3 4 5 6 7
Language to be used	•	•	•	1 2 3 4 5 6 7
Reliability	•	•	•	1 2 3 4 5 6 7
Project Risk	•	•	•	1 2 3 4 5 6 7
Development environment	•	•	•	1 2 3 4 5 6 7
Schedule constraints	•	•	•	1 2 3 4 5 6 7
No of users	•	•	•	1 2 3 4 5 6 7
Other (please list):				
______________________		•	•	1 2 3 4 5 6 7
______________________		•	•	1 2 3 4 5 6 7
______________________		•	•	1 2 3 4 5 6 7
______________________		•	•	1 2 3 4 5 6 7
______________________		•	•	1 2 3 4 5 6 7

D.9 For projects please indicate and rate your **normal** size, duration, effort and cost estimating practices.

Is the **SIZE** of a project estimated?

Yes No (If yes complete the following else go to the next section)

Cross the technique(s) that are **currently used** in the organisation.	Frequency of use. (1=never; 4=sometimes; 7=always)	Ease of use (1=poor; 4=average; 7=excellent)	Accuracy. (1=inaccurate, 4=accurate, 7=very accurate)
Analogy	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7
Expert Judgement	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7
Work Breakdown	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7
Algorithmic model eg. Function Points	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7
Other (please specify ______	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7

Is the **DURATION** of a project estimated?

Yes No (If yes complete the following else go to the next section)

Cross the technique(s) that are **currently used** in the organisation.	Frequency of use. (1=never; 4=sometimes; 7=always)	Ease of use. (1=poor, 4=average; 7=excellent)	Accuracy. (1=in[illegible]ate; 4=accurate; 7=[illegible]ccurate)
Analogy	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 [illegible] 6 7
Expert Judgement	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 [illegible]
Work Breakdown	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7
Algorithmic model	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7
Other (please specify ______	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7

Is the **EFFORT** of a project estimated?

Yes No (If yes complete the following else go to the next section)

Cross the technique(s) that are **currently used** in the organisation.	Frequency of use. (1=never; 4=sometimes; 7=always)	Ease of use. (1=poor; 4=average; 7=excellent)	Accuracy. (1=inaccurate, 4=accurate: 7=very accurate)
Analogy	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7
Expert Judgement	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7
Work Breakdown	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7
Algorithmic model	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7
Other (please specify ______	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7

Is the **COST** of a project estimated?

Yes No (If yes complete the following)

Cross the technique(s) that are **currently used** in the organisation.	Frequency of use. (1=never; 4=sometimes; 7=always)	Ease of use. (1=poor; 4=average; 7=excellent)	Accuracy. (1=inaccurate; 4=accurate; 7=very accurate)
Analogy	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7
Expert Judgement	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7
Work Breakdown	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7
Algorithmic model	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7
Other (please specify ______	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7

PART E: General Software Estimation Practices.

These questions relate to total project estimates and their revisions. (If you require more space to write your answers please use the back of this form).

E.1 What procedure does your organisation have for estimating?
No procedure   Informal procedure
Documented procedure

E.2 Do you use more than one technique to estimate a **single** project? (1=never; 4=sometimes; 7=always).

1 2 3 4 5 6 7

E.3 Do you keep records of project estimates? (1=never; 4=sometimes; 7=always).

1 2 3 4 5 6 7

E.4 Do you provide an indication of the probability of achievement for each estimate? (1=never; 4=sometimes; 7=always).

1 2 3 4 5 6 7

E.5 Do you produce a range of possible values for each estimate? (1=never; 4=sometimes; 7=always).

1 2 3 4 5 6 7

E.6 Are the estimates peer reviewed? (1=never; 4=sometimes; 7=always).

1 2 3 4 5 6 7

E.7 How often are estimates revised?
Never
Only as specifications change
At project review points
Constantly under review
Other (please note) ________________

E.8 How frequently do you compare your estimates with the actuals? (1=never; 4=sometimes; 7=always).

1 2 3 4 5 6 7

E.9 **How frequently do you use a formal feedback mechanism to improve future estimates? (1=never; 4=sometimes; 7=always).**

1 2 3 4 5 6 7

E.10 How important is software project estimating to your organisation? (1=not important; 4=important; 7=extremely important).

1 2 3 4 5 6 7

E.11 How satisfied are you with the estimation process within your organisation? (1=very dissatisfied; 4=okay; 7=very satisfied).

1 2 3 4 5 6 7

E.12 How accurate are the estimates made by the organisation. (1=inaccurate; 4=somewhat accurate; 7= very accurate)

1 2 3 4 5 6 7

E.13 Why is estimating undertaken in the organisation?

To provide firm quotations
Organisational requirement
To obtain project approval
Assess project risk
Budgeting
Other. (please specify):
____

E.14 In which phase of the project are you required to provide an **initial** estimate? (please specify):

____

E.15 Approximately how many total project estimates have been made in the **organisation** in the last year?

New Projects [ ]

Maintenance Projects [ ]

E.16 Have **you previously used** but **discarded** some estimating practices?

Yes No (If yes please complete the following else go to E17).

Cross the technique(s) **discarded**.	If the technique was automated please specify the software.	Please specify the technique & its source.	Please indicate the reason(s) why you stopped using the technique.
Analogy	______ ______ ______ ______ ______ ______ ______	Published in general literature name: ______ Proprietary name: ______ Developed in-house Developed by consultants	Lack of training Found to be inaccurate Too difficult to use Too time consuming Inappropriate for your projects Other (please specify): ______
Expert Judgement	______ ______ ______ ______ ______ ______ ______	Published in general literature name: ______ Proprietary name: ______ Developed in-house Developed by consultants	Lack of training Found to be inaccurate Too difficult to use Too time consuming Inappropriate for your projects Other (please specify): ______
Work Breakdown	______ ______ ______ ______ ______ ______ ______	Published in general literature name: ______ Proprietary name: ______ Developed in-house Developed by consultants	Lack of training Found to be inaccurate Too difficult to use Too time consuming Inappropriate for your projects Other (please specify): ______
Algorithmic model e.g. Function Points	______ ______ ______ ______ ______ ______ ______	Published in general literature name: ______ Proprietary name: ______ Developed in-house Developed by consultants	Lack of training Found to be inaccurate Too difficult to use Too time consuming Inappropriate for your projects Other (please specify): ______
Other (please specify) ______ ______ ______ ______ ______	______ ______ ______ ______ ______ ______ ______	Published in general literature name: ______ Proprietary name: ______ Developed in-house Developed by consultants	Lack of training Found to be inaccurate Too difficult to use Too time consuming Inappropriate for your projects Other (please specify): ______

E.17 Please indicate **your current normal** estimating practices. Also indicate if the practice is **normally used** in your **organisation**.

Cross the technique(s) that are **currently used.**	If the technique was automated please specify the software.	Please specify the technique & its source.	Please indicate the reason(s) why the technique is used.
Analogy Self Organisation	____________ ____________ ____________ ____________ ____________ ____________ ____________	Published in general literature name: ____________ Proprietary name: ____________ Developed in-house Developed by consultants	Training readily available Found to be accurate Easy to use Organisational standard Appropriate for the projects Other (please specify): ____________
Expert Judgement Self Organisation	____________ ____________ ____________ ____________ ____________ ____________ ____________	Published in general literature name: ____________ Proprietary name: ____________ Developed in-house Developed by consultants	Training readily available Found to be accurate Easy to use Organisational standard Appropriate for the projects Other (please specify): ____________
Work Breakdown Self Organisation	____________ ____________ ____________ ____________ ____________ ____________ ____________	Published in general literature name: ____________ Proprietary name: ____________ Developed in-house Developed by consultants	Training readily available Found to be accurate Easy to use Organisational standard Appropriate for the projects Other (please specify): ____________
Algorithmic model e.g. Function Points Self Organisation	____________ ____________ ____________ ____________ ____________ ____________ ____________	Published in general literature name: ____________ Proprietary name: ____________ Developed in-house Developed by consultants	Training readily available Found to be accurate Easy to use Organisational standard Appropriate for the projects Other (please specify): ____________
Other (please specify) ____________ ____ Self Organisation	____________ ____________ ____________ ____________ ____________ ____________ ____________	Published in general literature name: ____________ Proprietary name: ____________ Developed in-house Developed by consultants	Training readily available Found to be accurate Easy to use Organisational standard Appropriate for the projects Other (please specify): ____________

E.18 Please state your opinion on how you think the software intensive project estimating could be improved in your organisation?

E.19 Please state your opinion on how estimating techniques could be improved in the software industry.

E.20 Please advise of any other aspect of the organisation's estimating practice that has not been covered by this questionnaire.

Thank you for completing this questionnaire. Your time and effort is appreciated.

Appendix 2 - Letter to Respondents

28th October 1997

Information Technology Manager
«Company_Name»
«Address» «Postcode»

Dear Madam/Sir,

The School of Computing, Information and Mathematical Sciences at Edith Cowan University is active in Software Engineering teaching and research The Software Engineering research group's focus is on areas that have direct practical industrial application. One of these areas, software project estimating, is critical for planning and managing software projects.

As part of the work in this area we are conducting a survey to ascertain current practices and obtain your views on the strengths and weaknesses of existing techniques. Comprehensive data on current Australian practices in this area does not exist. Your views will assist us in gaining a greater understanding of current practice and also enable us to target our estimating research activities to better suit the industry.

While we realise that any extra demand on your time is an imposition we do need your help in having your most knowledgeable estimating people complete the attached questionnaire and return it in the reply paid envelope (Two questionnaires are enclosed but please feel free to copy as many as you need). Pilot studies have indicated the questionnaire will take 30 to 45 minutes to complete.

All information will be treated as confidential. However, if you would like a copy of the resulting report please complete your contact details on the questionnaire

We would be grateful for any assistance you can provide. If you have any enquiries please do not hesitate to call the undersigned on 08 9370 6363.

Yours sincerely

Stuart Hope
Software Engineering
School of Computing, Information and Mathematical Sciences