A TOTAL QUALITY MANAGEMENT (TQM) STRATEGIC MEASUREMENT PERSPECTIVE WITH SPECIFIC REFERENCE TO THE SOFTWARE INDUSTRY

by

MARTHA JACOBA POHL

submitted in accordance with the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in the subject

OPERATIONS RESEARCH

at the

UNIVERSITY OF SOUTH AFRICA

PROMOTER: PROF CA VAN DER MERWE

NOVEMBER 1996

ACKNOWLEDGEMENTS

Many people have provided me with incalculable aid in bringing this dissertation into existence. I would like to offer my heartfelt thanks to:

- My promoter, Professor Carel van der Merwe for his professional guidance, continual support and encouragement.
- Anet Coetzee, for her dedication and proficient document processing.
- Elsie Verheem, for walking that extra mile in library support.
- Dr B Kitchenham, for provision of MERMAID project data.
- My husband, Fred for editing the manuscript and ongoing love and support.
- My children, Maricke and Carissa for their spontaneous acts of love when most needed.
- My parents, Elie and Ella Nel for their unselfish support.

Above all, I wish to honour and thank the Lord.

EXECUTIVE SUMMARY

The dissertation aims to obtain an integrated and comprehensive perspective on measurement issues that play a strategic role in organisations that aim at continuous quality improvement through TQM.

The multidimensional definition of quality is proposed to view quality holistically. The definition is dynamic, thus dimensions are subject to evolution. Measurement of the quality dimensions is investigated. The relationship between quality and cost, productivity and profitability respectively is examined. The product quality dimensions are redefined for processes.

Measurement is a strategic component of TQM. Integration of financial measures with supplier-; customer-; performance- and internal process measurement is essential for synergism. Measurement of quality management is an additional strategic quality dimension. Applicable research was integrated. Quantitative structures used successfully in industry to achieve quality improvement is important, thus the quality management maturity grid, cleanroom software engineering, software factories, quality function deployment, benchmarking and the ISO 9000 standards are briefly described.

Software Metrics Programs are considered to be an application of a holistic measurement approach to quality. Two practical approaches are identified. A framework for initiating implementation is proposed.

Two strategic software measurement issues are reliability and cost estimation. Software reliability measurement and modelling are introduced. A strategic approach to software cost estimation is suggested. The critical role of data collection is emphasized. Different approaches to implement software cost estimation in organisations are proposed. A total installed cost template as the ultimate goal is envisaged. An overview of selected software cost estimation models is provided. Potential research areas are identified. The linearity/nonlinearity nature of the software production function is analysed. The synergy between software cost estimation models and project management techniques is investigated.

The quantification aspects of uncertainty in activity durations, pertaining to project scheduling, are discussed. Statistical distributions for activity durations are reviewed and compared. A structural view of criteria determining activity duration distribution selection is provided. Estimation issues are reviewed.

The integration of knowledge from dispersed fields leads to new dimensions of interaction. Research and practical experience regarding software metrics and software metrics programs can be successfully applied to address the measurement of strategic indicators in other industries.

KEYWORDS

Total Quality Management; quality dimensions; strategic measurement; quality management measurement; software metrics programs; software cost estimation; PERT; activity duration distributions; software reliability; project management techniques.

CONTENTS

CHAPTER 1 INTRODUCTION

CHAPTER 2 EVOLUTION OF THE QUALITY CONCEPT

CHAPTER 3 MEASUREMENT AND TOTAL QUALITY MANAGEMENT

CHAPTER 4 SOFTWARE METRICS PROGRAMS

CHAPTER 5 STRATEGIC MEASUREMENT ISSUES IN SOFTWARE

CHAPTER 6 QUANTIFICATION ASPECTS OF UNCERTAINTY IN ACTIVITY DURATIONS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

When you can measure what you are speaking about,
and express it in numbers, you know something about it;
when you cannot express it in numbers,
your knowledge is of a meagre and unsatisfactory kind;
it may be the beginning of knowledge,
but you scarcely in your thoughts advance
to the stage of science........

(Lord Kelvin 1889, [Conte, Dunsmore & Shen 1986])

Organisations in a changing South Africa are currently competing in an increasingly unstable and competitive environment. Top management need to keep track with change and can only do so by reviewing and renewing their organisational structures and processes and adopt new business techniques.

Quality, as a strategic variable, is considered to be one of the most important components for the survival, growth and competitive position of an organisation. Quality can be both a problem and an opportunity for companies. To pursue it as an opportunity a deeper understanding of its history, meaning, measurement and sources is needed.

Quantitative information regarding quality and all the components thereof, is becoming increasingly important for the top management decision-making process. Definitional inconsistencies and measurement difficulties have, however, prevented rigid quantitative studies.

This dissertation will describe the development of a strategic measurement perspective for organisations within the Total Quality Management (TQM) framework, with specific reference to the software industry.

A perspective is defined as the ***apparent relation between different aspects of a problem*** (Oxford Dictionary of Current English 1974, s.v. "perspective").

A strategic measurement perspective thus refers to the relation between the different measurement aspects that are of strategic importance in an organisation, within the Total Quality Management framework.

Total Quality Management (TQM) is defined as: ***both a philosophy and a set of guiding principles that represent the foundation for a continuously improving organisation. TQM is the application of quantitative methods and human resources to improve the material and services supplied to an organisation, and the degree to which the needs of the customer are met, now and in the future. TQM integrates fundamental management techniques, existing improvement efforts, and technical tools under a disciplined approach focused on continuous improvement*** (The American Department of Defence definition quoted in Schulmeyer & McManus 1992: xxxi).

A systematic, integrated and consistent organisation-wide perspective to examine the work processes is thus needed to improve quality comprehensively.

1.1 OUTLINE OF THE STUDY

The aim of the study is to develop a coherent view of the measurement aspects, within the quality drive, that are of strategic importance to an organisation.

As departure and anchor point, the evolution of the quality concept is discussed in chapter 2. Evolution is considered in this instance as meaning **the process of developing**. Quality was traditionally seen as a one dimensional concept and defined as conformance to specifications as embodied in the quality control/assurance concepts. Currently, quality is defined and interpreted in many ways. It has different meanings in different industries. These differences are a result of the existence of different approaches to quality. Transcendent-, user-, product- and manufacturing quality approaches exist (Garvin 1984). These approaches and their importance are discussed.

To acknowledge the different approaches, quality is viewed as a multi-dimensional entity (Garvin 1984). A multi-dimensional quality definition is proposed as the core concept to describe quality holistically. Garvin (1984) has identified eight critical dimensions - performance, features, reliability, conformance to specifications, durability, aesthetics, perceived quality and serviceability. These dimensions are described and the acknowledgement of the multidimensional nature of quality is discussed.

With time, the dimensions of quality will change and are added to as a result of changes in the nature of demand of products. Reasons for changes and three additional dimensions proposed by Van der Merwe (1989) - adaptability, destructibility and availability, are briefly described.

The measurement of the quality dimensions remains a difficult task. Some dimensions, such as reliability, are much easier to quantify than e.g. perceived quality. Determining and quantifying quality dimensions are usually product-related. Literature studies concerning quality dimensions, mainly emphasize and define the dimensions that relate to customer satisfaction/delight in a particular context, e.g. health care quality dimensions. A brief discussion of the above issues is given.

The importance of viewing quality multidimensionally becomes clear when one considers the strategic impact, particularly in relation to cost, productivity and profit. These aspects are not covered extensively but the important issues are summarised.

Redefining the product quality dimensions to that of process quality dimensions are a natural extension. The dimensions are defined and examples are provided.

Thus, viewing quality multi-dimensionally, enables one to put the complex role of quality in the business environment in perspective.

Chapter 3 aims to obtain a perspective on the strategic measurement issues within the TQM movement. One of the cornerstones of TQM is the requirement for continuous and accurate measurement for every process that exists within the organisation, i.e. an internal view (Barrier 1992). There is also worldwide recognition that the impact and effectiveness of Quality Programs need to be measured, i.e. an external evaluation view. Harari (1993) declares that one of the main reasons why TQM fails is the focus of TQM on internal processes rather than on external results.

The development of the discipline of quality, i.e. the period of inspection, then quality control, quality assurance and currently strategic quality management is described in the first section. Throughout, measurement has been, and still is, an integral part of the process of achieving quality.

The second section consists of a comprehensive and integrated discussion on internal and external measurement aspects that relate to quality in its strategic importance context. Aspects are:

1) the instrumental role of measurement in the link of quality to strategic and financial management
2) customer measurement (by looking at it as a component in the measuring of quality, not as an end product in itself)
3) performance, measurement and quality (highlighting the relation and interaction between quality and performance measurement)
4) supplier measurement approaches and supplier quality certification that are used by companies
5) quality and measurement systems
6) the key role of measurement in the Malcolm Baldrige National Quality Award for businesses in the United States of America.

The aspects are discussed in the broader perspective, namely looking at it from a strategic multidimensional business viewpoint and not from a statistical process control viewpoint. Although the latter is an integral part of most of the quality improvement processes, it will not be specifically described in this dissertation.

Only by integrating and linking of internal and external measurements of quality, businesses will achieve optimum benefits. The use of this information by the organisation in its pursuit of quality, needs to be part of the planning process in the development of measurement systems. Adequate definition, planning, process change, implementation and evaluation is extremely important. These aspects have not been addressed adequately. Godfrey (1993: 56) considers the aspect of data and information needs, as one of the ten areas of future research in TQM. He remarks: *few researchers have looked at the data and information needs of companies engaged in serious TQM efforts.*

An additional dimension in the measurement of quality at a strategic level, is the measurement of quality management in organisations. The third section covers the measurement of quality management. An instrument for measuring the critical factors of quality management, developed by Saraph, Benson & Schroeder (1989), as well as the use of this instrument to test the effect of organisational context on quality management by means of an empirical study (Benson, Saraph & Schroeder 1991), is described. A framework for quality management research and an associated measurement instrument suggested by Flynn, Schroeder & Sakakibara (1994) are also described and compared to the work of Saraph et al. (1989).

The fourth section describes innovative quantitative structures for process improvement currently used in industry as a vehicle to support, control and measure improvement. The quality management maturity grid, cleanroom software engineering, software factories, quality function deployment, the seven planning tools, benchmarking and the ISO 9000 series of standards are described in terms of what each constitutes.

An attempt to apply a holistic measurement approach to quality is software metrics programs. Software metrics programs, the name for organisation-wide measurement programs in the software industry, are discussed in chapter 4. The aim of developing software that is on time, within budget and of good quality has led many software organisations to adopt a software metrics program.

The role of a software metrics program, by measuring variables in each of the key areas that impact the organisation, is to identify strengths and weaknesses, pinpoint areas for improvement, make recommendations and provide follow-up measures to identify patterns over time. It thus represents a long-term management commitment to understand and manage software development better.

Chapter 4 begins with clarifying the definitional aspect of software metrics terms. The following implementation aspects of software metrics programs are then addressed:

1) organisational requirements
2) different measurement approaches (two approaches are identified: the global and the project-oriented approach. A table summarising the procedure for each approach as well as the advantages and disadvantages of each are provided.)
3) a practical framework which is proposed to plan and develop the process of metric collection that can be used with each of the above mentioned approaches
4) the critical role of accurate, on-time and sufficient data collection and the need for a company-wide database. The selection of a package for the database is also discussed.
5) measurement tools
6) the core role of the human in software metrics programs
7) training and consultation
8) implementation problems
9) evaluation and feedback.

The state of the practice of software metrics programs worldwide are summarised and the extension of the concept of the metric approach to other industries is investigated, specifically in relation to key performance indicators (KPI's).

The general reader is thus familiarized with the software metrics concept and software metrics programs in order to stimulate the possible use of such programs in other industries.

Quality, time and cost constitute the three dimensions of software development. Two strategic quantitative issues in the software industry that are closely interlinked with achieving the aim of software metrics programs, i.e. continuous improvement, are software reliability and software cost estimation. These two issues are the subjects of chapter 5.

Software reliability is a quantifiable dimension of quality. The impact of software failure as a result of poor reliability is large and can often be critical. The IEEE/ANSI[1] (Standard 982.2) definition is: ***Software reliability is the probability that software will not cause the failure of a system for a specified time under specified conditions*** (Pfleeger 1992: 57).

Software reliability is only described by means of a brief introduction to the subject. Definitions of terms that are important within the context of software reliability are given. Software reliability measures, effective control and evaluation mechanisms, and their applications are described. Software reliability modelling is defined and described. The section ends with a list of identified current research areas regarding reliability modelling.

A successful software development project is one that meets its cost, schedule and quality goals. An internationally recognised problem in software organisations is "overrun" in terms of budget and time schedules. Software Cost Estimation, defined as the empirical process of estimating effort and duration, and thus costs, is a serious problem for project management and is intrinsically linked to quality. Improved effectiveness of both effort- and duration estimation of software projects is therefore extremely important.

The following aspects regarding software cost estimation are addressed:

- the approach to software cost estimation
- definitions of relevant software metrics
- software cost estimation requirements
- software cost estimation models
- software cost estimation tools
- a software cost template.

A strategic approach (i.e. not prescribing the use of one technique or tool but recommending solutions for different aspects of the problem) is proposed for software cost estimation. The dynamic nature of software cost estimating is acknowledged and the critical role of data collection is emphasized.

[1] The 1988 IEEE Guide for the Use of IEEE Standard Dictionary of Measures to Produce Reliable Software.

It is suggested that either of two directions (or a hybrid of these) can be followed when implementing software cost estimation modelling in an organisation:

1) use an established model(s) but calibrate the model(s) for the specific environment or
2) develop a local cost estimation model by using the framework suggested in chapter 5.

The development of a total installed cost template (Wellman 1993) is envisaged as the ultimate goal.

Nine areas of current research interest in software cost estimation modelling are identified. One of these areas, regarding the assumption of a nonlinear relationship between size and effort in software cost estimation models, is currently a subject of controversy. Current published results (Banker, Chang & Kemerer 1994; Kitchenham 1992) are investigated and some preliminary research results are included in the dissertation.

In addition, the link between software cost estimation and project management techniques is investigated. Current knowledge is integrated, a comparison is made between estimating and project management tools and seven areas for research identified. One of those, the quantification of uncertainty in activity durations, will be the subject of chapter 6.

Chapter 6 describes the quantification aspects of uncertainty in activity (task-time) durations for project scheduling purposes. The management of projects and its ultimate success/failure will largely depend on the quality of the planning of the project. Good project planning is thus of strategic importance to an organisation. It constitutes a key success factor. A crucial aspect of project planning is project scheduling. To determine the risks involved, the quantification of uncertainty in activity duration is needed. It is thus a strategic measurement issue. It will ultimately influence the quality of the end product because of schedule compression if not properly addressed. Only uncertainty of activity durations within activity networks are discussed.

The chapter aims to:

1) supply a structured view of the criteria that determine the selection of an activity duration distribution
2) review and compare the suggested statistical distributions for activity durations
3) integrate current knowledge on estimation issues relating to activity durations and to suggest research regarding the project completion time distribution when using the "distribution-free" approximations for the mean and variance of activity durations.

2. EVOLUTION OF THE QUALITY CONCEPT

The Caterpillar and Alice looked at each other for some time in silence:
at last the Caterpillar took the hookah out of its mouth, and addressed her in a languid, sleepy voice.
"Who are you?" said the Caterpillar.
This was not an encouraging opening for a conversation. Alice replied, rather shyly, "I - I hardly know, Sir, just at present - at least I know who I was when I got up this morning, but I think I must have been changed several times since then."
"What do you mean by that?" said the Caterpillar sternly. "Explain yourself!"
"I can't explain myself, I'm afraid, Sir," said Alice, "because I'm not myself, you see."
"I don't see," said the Caterpillar.
"I'm afraid I can't put it more clearly," Alice replied very politely, "for I can't understand it myself to begin with; and being so many different sizes in a day is very confusing."

Alice in Wonderland (Lewis Carroll 1865)

The above scene from the story depicts the same type of confusion that exists about the concept of quality, as well as the evolutionary nature of quality.

This chapter will deal with the evolution of the product quality concept. Evolution is defined in this context as ***the process of developing***. Quality was traditionally seen as a one dimensional concept and defined as ***conformance to specifications*** as embodied in the quality control/assurance concepts. Quality is currently defined and interpreted in many ways. It has different meanings in different industries. These differences are the result of the existence of different approaches to quality. Transcendent-, user-, product- and manufacturing approaches exist (Garvin 1984). To acknowledge the different approaches, quality is viewed as a multi-dimensional entity (Garvin 1984). The multidimensional definition of quality is proposed as the core concept in viewing quality holistically. With time, dimensions change and are added to.

The different approaches to quality, the multidimensionality and evolutionary nature of the dimensions of quality will be discussed. In addition, the aspect of measurement of the quality dimensions as well as the strategic impact of the quality dimensions on business performance, particularly cost, profit and productivity is summarised. The product quality dimensions are redefined for process quality.

2.1 APPROACHES TO QUALITY

Different approaches to quality exist. Garvin (1984) discusses the transcendent-, user-, product-, value-, and manufacturing-based approaches to quality. Each one of the approaches is briefly discussed.

The transcendent approach: According to this approach, quality cannot be defined precisely. It is a property that we learn to recognise only through experience and is not analysable (Garvin 1984). Smith (1993) states that this approach does not facilitate measurement efforts, but does reflect the concept's meaning. In his article "The meaning of quality", Smith (1993) presents a conceptual analysis of quality.

Smith (1993) declares quality a property term or attribute as it refers to a characteristic of some object. It can not be conceived as existing apart from its object. He further notes that quality is not directly measurable. He regards quality as an abstract characteristic. Determining the quality of an object may involve taking measurements of many of it's attributes. This agrees with the view of Garvin (1984) who terms it quality dimensions and Ishikawa (1990) who terms it quality characteristics. However, Smith (1993) regards these measurements as surrogate measures of quality, but not measures of quality itself. He also defines quality as a relational attribute. Such an attribute applies to an entity but characterizes it only in relationship to something else. According to Smith (1993), quality indicates the relationship between certain of the entity's attributes - its "quality characteristics" - and an evaluative standard or criterion. The standards can be approximately objective for the kind of entity in question, reflecting the ideal prototype which people mentally conceive for such things. He cites the example of the Malcolm Baldrige National Quality Award or the ISO 9000 standards as criteria for assessing the quality of an organisation's quality management activities. It can also be that the evaluative standard represents the interest, needs, preferences or values of an individual or group.

He further argues that the assessment of quality is a judgmental process. Quality assessment entails determining user needs, identifying entity attributes or quality characteristics which relates to those needs, assessing the entity's merit on each of the attributes, and consolidating these partial scores into a final judgment of quality. Quality is thus subjective, assessed from a certain perspective, reflecting the standard used as a criterion.

Smith (1993: 237) proposes the following definition for quality: ***Quality is the goodness or excellence of something. It is assessed against accepted standards of merit for such things and against the interests/needs of users and other stakeholders.***

The user-based approach: The approach is a personal view of quality and is subjective. According to Smith (1993), a shift to user-based definitions of quality has been noticed with the growing acceptance of TQM in business. It is the dominant current approach to quality. Smith (1993) stresses that most quality assessments are currently specified in terms of the needs of an object's users, with the majority of these involving consumer evaluations of products-for-sale. Juran's phrase "fitness for use" is a very apt description of this view of quality.

Two problems with this approach (Garvin 1984) are the following:

1) the aggregation of varying individual preferences so that they lead to meaningful definitions of quality at the market level and
2) the distinguishing of those product attributes that connote quality from those that simply maximize customer satisfaction.

The instrument, SERVQUAL, which measures service quality dimensions (Parasuraman, Zeithaml

& Berry 1988) is a step in the direction of addressing the second problem. It will be briefly described in 2.4 below.

The main problem, according to Smith (1993), is operationalization. Difficulty arises in determining user needs and translating user needs into specific attributes (a problem addressed by Quality Function Deployment which will be described in chapter 3.4). He stresses that product quality can thus not be equated with user needs. However, he states that this conceptualization is the most influential in current quality research and practise.

The product-based approach: The product-based approach defines quality as a precise and measurable variable. According to Garvin (1984): *Differences in quality relates differences in the quantity of some ingredient or attribute possessed by a product. It lends a vertical or hierarchical dimension to quality, for goods can be ranked according to the amount of the desired attribute that they possess.* A problem with this approach is that unambiguous ranking is only possible if the attributes in question are considered as preferable by all buyers.

Two corollaries to this approach is:

1) higher quality can only be obtained at higher cost and
2) quality is viewed as an inherent characteristic of goods (Garvin 1984).

This leads to the view that quality can be assessed objectively, and is based on more than preferences alone.

Smith (1993) states that product-based definitions fail to acknowledge the relational nature of quality, i.e. its dependence on an outside standard or stakeholder.

The value-based approach: The value-based approach defines quality in terms of costs and prices. A quality product is one that delivers performance at an acceptable price, or conformance at an acceptable cost (Garvin 1984). The difficulty in applying this approach lies in the blending of two related but distinct concepts. Quality is equated with value, resulting in a hybrid "affordable excellence" (Garvin 1984). It lacks well-defined limits and is difficult to apply in practise.

The manufacturing-based approach: The manufacturing approach is mainly used within engineering and manufacturing practises. Quality is defined as conformance to specifications (Garvin 1984). The primary focus of this approach is internal quality control and it is not customer-based. This approach has placed emphasis on reliability engineering and statistical quality control, which both aim at cost reduction.

According to Smith (1993), the adequacy of product specifications as quality standards is questionable. He added that specifications define a product that will perform its intended function and will have no real merit or significance beyond that.

Smith (1993) states that user needs is the primary quality criterion for a consumer product, with design specifications an operational surrogate. He concludes that when product design reflects a comprehensive understanding of user needs, specifications can be an appropriate criterion for product quality. If specifications are developed without knowledge of user needs and achieve "bare-boned" product functionality, they are an inadequate standard.

2.1.1 IMPORTANCE OF THE QUALITY APPROACHES

According to Garvin (1984), the coexistence of the different approaches has important implications and must be acknowledged. It helps to clear the often competing views of quality. A single definition of quality is a frequent source of disagreement. However, Perry (1992) warns that the approaches often conflict or overlap, and may lead to disparate conclusions.

Garvin (1984) advises that the approach to quality needs to shift as one moves from the design to the marketing of a product. The characteristics that connote quality must first be identified through market research (user-based), these characteristics must then be translated into identifiable product attributes (product-based) and the manufacturing process must then be organized to ensure that products are made precisely to these specifications (manufacturing-based). A process that ignores any one of these steps will not result in a quality product. All three views are necessary and should be cultivated.

The Quality Function Deployment technique or as it is also known, The House of Quality, is a technique that combines the above-mentioned approaches to address quality and is described in chapter 3.4.4.

Smith (1993) challenges Garvin's view on shifting one's approach to quality throughout the business process, advising that management must at all times consider its products from both a consumer and producer perspective, ensuring that they satisfy user needs as well as being profitable or otherwise beneficial to the firm.

Parasuraman et al. (1988) acknowledges the approaches and makes a distinction between objective quality (product-based and manufacturing-based approach) and perceived quality (user-based approach) which he uses in developing the SERVQUAL instrument (to measure service quality).

Forker (1991: 70) summarises the five prominent quality theoreticians' approaches to quality and the major focus of each of their definitions in table 2.1 as follows:

EXPERT	APPROACH	MAJOR FOCUS OF QUALITY DEFINITION
DEMING	USER-BASED	HOW WELL A GOOD OR SERVICE MEETS CONSUMER'S NEEDS
JURAN	USER-BASED	FITNESS FOR USE
CROSBY	MANUFACTURING-BASED	CONFORMANCE TO REQUIREMENTS
TAGUCHI	VALUE-BASED	OPERATION OF PRODUCT IN INTENDED MANNER WITHOUT VARIABILITY
L'VOV	PRODUCT-BASED	TOTALITY OF A PRODUCT'S PROPERTIES WHICH DETERMINE ITS USEFULNESS

Table 2.1 Summary of Various Approaches to Quality

Five principles that are common to the quality approaches of Deming, Juran, Crosby and other authors on quality are given by Klaber (1993):

1) Definition of quality from the customer's point of view.
2) The practise of continuous improvement.
3) Act on data, facts and analysis.
4) The development of a [illegible]ng leadership team.
5) The making of an organisation-wide commitment to quality.

According to Quigley and McNamara (1992), Taguchi's loss function provides a vehicle for evaluating the user-, value-, manufacturing- and product approaches or "dimensions" as they call them. They advise purchasing departments in organisations to use the Taguchi loss concept as a method to evaluate the quality differences between suppliers by determining the value of the quality differentials. The buyer can calculate the total cost associated with the product that competing suppliers offer by combining value pricing and the Taguchi loss concept. The user-, value- and manufacturing "dimensions" are involved.

Smith (1993) criticizes Garvin's approaches to quality as follows:

1) He dismisses Garvin's product-based definition as inadequate since it fails to recognise the relational nature of quality.
2) He regards Garvin's definition of the user- and manufacturing based definition as valuable, but incomplete accounts for quality.
3) He argues that Garvin acknowledged the transcendent approach but did not say much about its definition. According to Smith, notions like goodness and excellence express the core meaning of quality.
4) He argues that Garvin's value-based definition is misconceived, in that price is conceptually distinct from product quality.

Smith (1993: 240) describes the current conceptualization of quality as *the consumer's evaluation of a product's fitness for use*. He argues that this notion does not fully express the concept's meaning. Quality is a property that can be ascribed to any entity, not just products-for-sale. Furthermore, quality can be assessed in terms of various standards and stakeholder perspectives, not just those of product users/consumers. He emphasizes that quality has become restricted to and equated with the term's meaning in its most important application, i.e. consumer evaluation.

Smith (1993: 241) wants to define quality as it relates to managerial and organisational affairs, in other words, quality for the purpose of TQM. He proposes the following definition of quality for the purpose of TQM: ***Quality is the goodness or excellence of any product, process, structure or other thing that an organization consists of or creates. It is assessed against accepted standards of merit for such things and against the interest/needs of producers, consumers and other stakeholders.***

He argues that not only the user- but also the producer-side view of quality is important to TQM. His proposed definition recognises this, avoiding serious failings of the consumer-side conceptualization.

He defines a producer-side view of quality as encompassing anything that makes a product valuable to its producer and not only a manufacturing or specification-based view in which producers develop specifications as the standard of excellence for a product that consumers are presumed to want.

Smith (1993) argues that the producers' and other stakeholders' views complement, but do not replace, prevailing consumer-side notions of quality. It clarifies and helps to resolve the intra-organisational conflicts that often arise over issues of quality. Manufacturing assesses product quality from the producer point of view, whereas marketing adopts the consumer's perspective. Both views are legitimate, thus judgmental trade-offs must be made in determining what is best for the firm.

Concerning products for sale, producer-side quality is primarily a matter of profitability: the firm's best products are those which are most profitable. Product profitability is largely driven by the costs of developing, producing, marketing and servicing the product. It is conceptually legitimate to consider what a producer values about its products, and to regard these attributes as comprising product quality from the producer's perspective (Smith 1993).

Smith (1993) concludes: *Organizations require a balanced approach to quality, one which considers their interests and the needs of their customers, as well as the legitimate concerns of other societal stakeholders. The proposed conceptualization, with it's explicit recognition of* *producer and other stakeholder views, provides such a balanced, sustainable perspective. It also encourages organisation members to regard all aspects of the organisation - what it creates and what it consists of - as opportunities for improvement, things that can be made excellent.*

The literature thus suggests that the different approaches are acknowledged and are used in determining instruments for measuring quality.

2.2 MULTIDIMENSIONALITY

Viewing quality multidimensionally, encapsulates the different approaches. Garvin (1984) identified the following critical dimensions: performance, features, reliability, conformance to specifications, durability, aesthetics, perceived quality and serviceability. Each one is self contained and distinct, as a product can be ranked high on one dimension whilst being low on another.

A short summary of each of the dimensions from Garvin (1984) is given.

1. **Performance**

   It refers to the primary operating characteristics of a product. It combines elements of both the product- and user-based approach. Measurable product attributes are used. Different brands can usually be ranked objectively on at least one dimension of performance. Thus, the performance of a product corresponds to its objective characteristics, whilst the relationship between performance and quality would reflect individual reactions.

2. **Features**

   Features are the secondary characteristics that supplement the product's basic functioning. It involves objective and measurable attributes; their translation into quality differences is equally affected by individual preferences. The distinction between the two is primarily one of centrality or degree of importance to the user.

3. **Reliability**

   It reflects the probability of a product's failing within a specified period of time. Common measures are mean time to first failure (MTFF), mean time between failures (MTBF), and the failure rate per unit time. This measure is more relevant to durable goods than to products and services that are consumed instantly. Japanese manufacturers have paid great attention to this dimension and obtained a competitive edge in several industries.

4. **Conformance**

It is the degree to which a product's design and operating characteristics conform to pre-established standards. Internal and external elements are involved. Internally, conformance is usually measured by the incidence of defects: the proportion of all units that fail to meet specifications, and thus require rework or repair. Externally, data is often difficult to obtain. Two common measures are the incidence of service calls for a product and the frequency of repairs under warranty. These measures neglect other deviations from the standard. More comprehensive measures of conformance are required if this aspect is to be utilised.

Both reliability and conformance (closely linked to the manufacturing approach) are relatively objective measures of quality, and are less likely to reflect individual preferences than are rankings based on performance or features.

5. **Durability**

It is a measure of product life and has both economic and technical dimensions. Technically, it can be defined as the amount of use one gets from a product before it physically deteriorates. It becomes difficult when repairs to a product is possible. The concept then takes on added dimensions, for product life will vary with changing economic conditions. Durability then becomes the amount of use one gets from a product before it breaks down and replacement is regarded as preferable to continued repair. This suggests that durability and reliability are closely linked. Durability figures should be interpreted with care as other social and economic factors, e.g. the use of longer-lived materials can be responsible for an increase in durability and not necessarily higher quality.

6. **Serviceability**

This is defined as the speed, courtesy and competence of repair. Some of these variables can be measured objectively; others reflect differing personal standards of what constitutes acceptable service. Responsiveness, one of these aspects, can be measured by the mean time to repair, while technical competence is reflected in the incidence of multiple service calls required to correct a single problem.

7. **Aesthetics**

This is a subjective measure as it involves how a product looks, feels, tastes, sounds or smells - a clear matter of personal judgement and reflection of individual preferences. The notion of ideal points in marketing was developed to capture this dimension of quality.

8. **Perceived quality**

Perceptions of quality is also a subjective assessment. It concentrates on aspects such as advertisements, image and brand names. It is defined as ***an abstract evaluation or judgement of a product that is formed from intrinsic attributes of the product (e.g. physical characteristics) and extrinsic attributes that are not part of the actual physical probduct (e.g. price, brand name, packaging)*** (Zeithaml 1987: iii).

2.2.1 SUMMARY AND CONCLUSION

The approaches to quality and the different quality dimensions can be related through the fact that each of the approaches focuses on a different dimension of quality. The product-based approach focuses on performance, features and durability, the user-based approach focuses on aesthetics and perceived quality; and the manufacturing-based approach focuses on conformance and reliability (Garvin 1984). If each dimension is considered separately, the sources of disagreement regarding the quality definition in the literature, becomes clear.

Currently, the multidimensional nature of quality is acknowledged in the literature as well as in the business world. Quality dimensions are defined and are usually related to the field under discussion, e.g. health care quality dimensions, service quality dimensions and software quality dimensions.

The different quality awards such as the Malcolm-Baldrige National Quality Award in the USA and the Deming Prize in Japan take the different dimensions into account when evaluating companies for the awards. The Malcolm-Baldrige Award will be described in chapter 3.2.8.

2.3 THE EVOLUTION OF THE QUALITY DIMENSIONS

Quality is an evolutionary concept. The changing pattern in the dimensions of quality happens because the nature of demand of products changes, probably because of:

1) **The rate of inflation.** Customers are more aware of the durability and reliability of products.

2) **Energy costs.** There is a shift towards energy-efficient goods and services as costs rise.

3) **Rising maintenance and repair costs.** High maintenance and repair cost related to a specific product may influence the less serious buyer in looking for an alternative product.

4) **Awareness of the eco-system.** Products need to adhere to strict environmental controls and new products are developed to be environment-friendly. This has changed the design, manufacturing and marketing aspects of products.

5) **Information technology.** Rapid development in this area has lead to new opportunities in design, manufacturing and marketing of products.

6) **Human issues.** Issues such as safety and health regulations, regarding the manufacturing as well as the consumption of the product by humans, change frequently as research results become available.

7) **Development of a global economy.** Information technology as well as political change has transformed the world into an environment for global competition. This has motivated companies even more to promote quality as the only weapon to stay competitive.

8) **The industrial emphasis on quality.** This has created an awareness of the concept and an attempt towards understanding it.

9) **The increasing ability of business to produce higher quality goods and services.** This has led to the consumer always wanting a "better", "smaller" or "different" product.

10) **Consumerism.** This is defined as the ***protection of consumers' interest*** (Oxford Universal Dictionary 1981, s.v. "consumerism"). Organisations as well as programs on television and radio are well established to protect customers and to fight for better quality products and services.

Current additional proposed dimensions are adaptability, destructibility (environment-friendliness) and availability.

ADAPTABILITY

Adaptability refers to the ability of a product to be used in different circumstances, e.g. environmental and changing technology constraints (Van der Merwe 1989).

DESTRUCTIBILITY

Destructibility refers to aspects such as pollution aspects and recycling. It is of particular importance if dangerous raw materials are used (Van der Merwe 1989). This dimension ties in with the concept of environment-friendly products, where bio-degradability "measures" environment-friendliness.

AVAILABILITY

Availability or shelf life refers to how the life span and durability of a product are influenced by storage as well as immediate availability at customer request (Van der Merwe 1989).

2.4 MEASUREMENT AND USE OF THE QUALITY DIMENSION CONCEPT

No reported measure of quality that captures the multiple dimensions suggested by Garvin (1984) exists yet (Karnes, Sridharan & Kanet 1995).

The measurement of quality for a product or service with regard to all the dimensions, is closely linked to the particular product or attribute in question. Indicators for each dimension differ naturally for different products, e.g. a car or a software product or service by a receptionist. Transferable uniform metrics (to measure the dimensions) do not exist across all business concerns.

Determination of the dimensions is an empirical task that has traditionally been addressed by market research (Smith 1993). Smith (1993) declares that this is consistent with his claim that quality is an abstract characteristic encompassing a variety of physical and non-physical attributes.

Literature studies, concerning quality dimensions and the measurement thereof, tend to emphasize the dimensions that relate to customer satisfaction and delight. This aspect has also been emphasized by Smith (1993), who warns that the current conceptualization of quality as *the consumer's evaluation of a product's fitness for use* means that quality has become restricted to and equated with the term's meaning in its most important application. For example, dimensions singled out in a study (Mowen, Licata & McPhail 1993) on service quality in medical care revealed trust, responsiveness and staff service as significant predictors of customer satisfaction. Mowen et al. (1993) conclude that situational context of the service may influence the quality dimensions that most affect consumer satisfaction.

Godfrey (1993) mentions the example of Banc One, the second most profitable bank in the world. They have established and measured the group of dimensions that addresses customer delight and have developed several statistical models to understand customer behaviour as a function of customer satisfaction. They found that delighted customers are five times as likely to buy other financial products from the bank as customers who are merely satisfied. These customers are also four times less likely to leave the bank than those who are just satisfied. They have also discovered that there is very little difference between customers who are satisfied and those who are neutral or even dissatisfied.

Urban (1993) describes steps taken by the Toronto Dominion Bank in an effort to deliver the quality dimensions of speed, accuracy and reliability of transactions it's customers want.

The most widely known current model of measuring service quality is the SERVQUAL instrument developed by Parasuraman et al. (1988). It assesses customer perceptions of service quality in service and retailing organisations. It thus measured the perceived quality dimension. Their research supports the notion that service quality is an overall evaluation similar to attitude. They separated perceived quality and satisfaction. Perceived quality is a global judgment or attitude, relating to the superiority of the service, whereas satisfaction is related to a specific situation. They view perceived service quality as the degree and direction of discrepancy between consumer's perceptions and expectations. Research by Parasuraman, Zeithaml & Berry (1985) found the service quality dimensions to be: Tangibles, Reliability, Responsiveness, Communication, Credibility, Security, Competence, Courtesy, Understanding/Knowing the customer and Access. Furthermore, as a service organisation differs from a manufacturing concern, features such as intangibility, heterogeneity and inseparability of production and consumption are important.

In measuring the quality dimensions clear distinction thus needs to be made with regard to whether one measures customer satisfaction/delight or the global quality dimensions.

Perry (1992) has done a survey to determine to what extent the dimensions listed by Garvin (1984) can be isolated and applied reasonably in the acquisition decision-making process. The objectives of the survey were:

1) *To identify, rank, and evaluate the dimensions of quality suggested by Garvin.*

2) *To determine the feasibility of applying these quality dimensions to the systems acquisition process.*

3) *To evaluate the quality feedback loop and the effectiveness of equipment warranties and* *other evaluation systems to measure or identify quality actually received* (Perry 1992: 19).

Perry (1992) applies an adaption of Garvin's dimensions to an industrial environment in the survey. He stresses that the data analysed were reported and not observed. Respondents were asked to respond to questions concerning their attitudes and actions instead of looking at these actions and their results from an established data collection source. His results imply that the identification of specific quality factors is feasible, that these factors can be evaluated in the systems acquisition process and in assessing product quality received, and that performance, reliability, durability and serviceability rank as the most important factors in most system acquisitions. Perry (1987) has also developed an analytical model for decision-making in the acquisition of capital equipment which considers the quality factors of performance, reliability, durability and serviceability. Perry (1992: 22) concludes: *The concept provides the buyer with a workable vehicle to bring together selected quality dimensions in the decision process in a cohesive and consistent manner that properly recognizes the inherent trade-off possibilities.* He, however, warns that it is only a tool and as such, the professional judgement of the buyer remains of utmost importance.

Karnes et al. (1995) incorporate the eight quality dimensions suggested by Garvin (1984) to measure quality from the consumer's perspective. They use the Analytical Hierarchy Process (Saaty 1980), a pairwise comparison approach, as the technique to measure overall quality.

The development of a generic framework for the measurement of the quality dimensions is considered as an important topic for further research in this area.

2.5 THE STRATEGIC IMPORTANCE OF THE QUALITY DIMENSIONS

Garvin (1984) stresses that the dimensions are not only of theoretical importance but are the key to use quality as a competitive weapon. He argues that attention should be focused on the separate dimensions of quality; markets must be closely examined for any untapped quality niches, and the organization must be tailored to support the desired focus. A few dimensions can be singled out for special attention. The selection of a defensible niche, however, is only a first step. Operational requirements must also be met, for each dimension of quality imposes its own demands on the firm (Garvin 1984).

The quality dimensions can also assist in the quantification of the cost and benefits of quality. Andreou (1991) argues that the impact of an investment in quality can be traced along each dimension and a clearer understanding can be obtained regarding possible interactions and trade-offs. Different strategic investment options can then be investigated.

Three business performance indicators: cost, productivity and profitability will be briefly discussed in relation to the quality dimensions.

2.5.1 COST

Garvin (1984) mentions the existence of three categories of theoretical discussions on the relationship between quality and cost.

Firstly, based on the product-approach, quality and direct costs are positively related. The implicit assumption is that quality differences reflect variations in performance, features, durability, or other product attributes that require additional commitment to resources.

Secondly, quality is seen as inversely related to cost. The costs of improving quality are argued to be less than the resulting savings in rework, scrap and warranty expenses. The practical measures that are employed include expenditures on:

1) prevention (e.g. quality planning, worker training and supplier education)
2) appraisal (e.g. product inspection and testing)
3) internal failures (e.g. rework and scrap)
4) external failures (e.g. warranty and product liability).

Thirdly, a number of analysts have extended the second category and claim that improved conformance should eventually lead to a reduction in long-term manufacturing costs.

Most empirical work (Garvin 1984) suggest that superior conformance (where conformance (a dimension) is used as a measure for quality) and total quality costs are inversely related. However, varying results obtained from studies reflect differences in the definitions of quality, i.e different dimensions are used, by firms in different industries. The PIMS (Profit Impact of Marketing Strategy) database which defines quality as an index (Andreou 1991; Garvin 1984) is a highly aggregated measure, thus different industries could have employed different definitions when assessing the quality of their products.

Maani (1988) indicates that a key issue in the debate on the cost of quality is the degree of reduction in costs as a result of improved quality. According to Maani (1988), Deming and Crosby maintain that the lowest quality costs can be achieved at the zero-defect level while Juran believes that the optimum costs of quality occurs at a non-zero level of defects. Juran argues that the preventative efforts for defect reduction have a diminishing rate of return which results in unproportionally higher marginal costs for eradication of the last few defects.

Again, the debate arises as a result of the different approaches (and thus different dimensions of quality) of each expert to quality.

Smith (1993) explains his view on the relationship between quality and product cost/price as follows: Cost is a key quality characteristic in the producer-side view owing to its impact on profitability. He emphasizes that price is by no means an aspect of the product's quality. One acquires the product and its quality characteristics in exchange for its price.

By acknowledging the producer-side view of quality, firms will not produce top-quality, high-price products for which there are no demand. Quality products, from a producer perspective, only include costs that yield corresponding quality benefits to consumers, which the latter are willing to pay for (Smith 1993).

Smith (1993) mentions that cost reduction, from a producer's perspective, increases product profitability, thus improving the product's quality for the firm.

Quality costs have traditionally been subdivided into three categories (Maani 1988):

1) prevention costs
2) detection costs
3) failure costs.

Prevention costs include such elements such as Quality Assurance programmes, design reviews, worker and supplier training, preventative maintenance, and purchasing and process improvement.

Detection or appraisal costs include inspection, sampling and testing performed at the input, output and in-process phases of manufacturing.

Failure costs consist of internal and external failures resulting in rejects, scrap, rework, service and warranty, and liability claims.

The literature suggests that in better performing companies, the breakdown of total quality costs are approximately 40, 25 and 35 percents for prevention, detection and failure whereas in poorly performing companies the percentages are expected to be in the vicinity of 5, 25 and 70 respectively (Maani 1988).

Total cost of quality, which include expenditure on prevention and inspection as well as the usual failure cost of rework, scrap and warranties, was found to be lower (less than one-half) at Japanese producers than the failure cost of that of the best USA companies (Garvin 1983).

According to Andreou (1991), quality is measured in an organization primarily through the management accounting system and the operating control system. The management accounting system measures the cost of rework, scrap and warranties. The operating control system uses statistical measures of quality, such as reject rates, customer returns and complaints, (again measurement of some of the dimensions of quality) which are not usually converted to financial measures. A survey conducted among industrial firms by CAM-I and the National Association of Accountants in the United States of America revealed that quality indicators are measured primarily through the operating control system (Andreou 1991). This type of information does not provide the level of detail needed for effective decision making. Quality and measurement systems will be discussed in chapter 3.2.7.

Andreou (1991) suggests the use of "Activity Based Costing" (abbreviated as ABC) as a technique to use in the strategic planning for quality. The central idea of ABC is to trace cost to products more accurately. A critical concept of ABC is that of a "cost driver", defined as any activity that results in cost being incurred. The cost driver measures the level of activity, e.g. the number of repairs required within a given period. The cost of the activity thus corresponds to the total cost of repairs required within a given period of time (Andreou 1991). By focusing on the cost drivers that are seen as quality cost drivers, a possible reduction in cost is possible. Typical quality cost drivers include: *product specifications (tolerances), process capabilities and limitations, procurement quality, product producibility, manufacturing systems and procedures, human error and variability, ... tooling, schedule stability and inspection* (Andreou 1991: 419).

By combining ABC principles with the concept of the "Value Chain" (a systematic display of basic activities involved in making a product), the capability to quantify the impact of quality improvement on the cost structure can be revealed (Andreou 1991).

Taguchi's loss function approach is currently advocated in the literature and used in practice to measure hidden quality costs for any variation of the actual value from the target value of a designated characteristic of a product (Kim & Liao 1994).

A recent book by Dale and Plunkett (1991) called "Quality Costing" gives a complete picture of the aspect of quality costing. They discuss aspects such as definitions of quality costing, collection of quality cost, reporting of quality cost, the use of quality cost, the setting up of a quality costing system and also present four case studies.

It is thus apparent that the quality dimensions are of strategic importance in quantifying cost.

2.5.2 PRODUCTIVITY

Quality and productivity are often seen as conflicting objectives, but the emerging view is that the two can be harmonious. Evidence and assertions support both views (Maani 1988). Stability and continuity in a manufacturing process are considered prerequisites by Hayes (1981) for increased productivity and improved quality. Maani (1988) suggests that it is important to identify the situations and conditions where a positive or inverse link between the two variables is likely to be present. These two possibilities will now be discussed.

2.5.2.1 Positive links

Maani (1988) indicates that most recent studies point to a positive (direct) link between quality improvements and productivity gains. Garvin (1983) observed, in a study on manufacturers of room air conditioners, that the strong relationship between quality and productivity is not explained by differences in technology and capital-intensive programs only. Companies with the highest quality were five times as productive (measured by direct labour assembly hours per unit) than companies with the poorest quality (Maani 1988). They had similar technologies and comparable capital-intensity. Evidence thus exists to indicate that better manufacturing-based quality results in higher output without a corresponding increase in cost (Maani 1988). Maani (1988) notes that the harmony between quality and productivity becomes evident when they are both seen as waste-free operations. If productivity is regarded as the ratio of defect-free output over inputs, then the positive relationship between quality and productivity becomes apparent (Maani 1988). The common practise of compromising quality to meet production schedules may then be abandoned in favour of the long-term competitive advantage of the firm.

Leonard and Sasser (1982) point out that quality and productivity can both be improved if managers are willing to make system changes to their operations and not only changing minor detail. Managers need to establish a new relationship between quality and cost as discussed.

2.5.2.2 Negative links

The discrepancy between definitions of productivity and quality are a possible explanation for the existence of negative links.

The negative relationship is usually present in operator-controlled tasks where an increase in productivity beyond a certain level would result in a sharp decline in quality. This can possibly explain why service industries which are characteristically labour intensive are generally less productive (Maani 1988).

Another case where a negative relationship exists is where a process or technology constrains productivity, i.e. where higher quality corresponds with lower productivity (Maani 1988).

The degree of labour and automation intensities could be a critical factor in determining the direction and extent of association between product quality and manufacturing productivity and is a potential area for further research (Maani 1988).

It is clear that the definition of quality (and thus once again the specific dimension(s) that are used) will influence the relationship that is established.

2.5.3 PROFITABILITY

Traditionally management regarded profit as their main responsibility. On achieving maximised profit for a certain level of investment, they argue that there is no incentive to improve quality as this will only lead to additional costs that will lower the profit. They believed that quality is to be run by a Quality Assurance Department. However, profit cannot really be maximised if a customer found the product to be of lesser quality and chose another product from a competitor.

The impact of quality on profit is usually not calculated due to limitations in traditional financial methods. Profit is usually measured by return on investment (Andreou 1991; Maani 1988). Empirical results point to a relationship between quality, profit and market share. However, most previous studies have used the PIMS (Profit Impact of Marketing Strategy) database which

1) defines quality as an index (highly aggregated and subjective measure) and
2) uses cross-sectional data (average performance of a company over a period of four to eight years).

Wagner (1984) analysed the PIMS data by using a time-series approach. His results indicated that improved return on investment is not necessarily the outcome for businesses that have or attained superior quality.

The relationship between profit and quality can be explained either via the market share path or the cost path as depicted in figure 2.1 (Garvin 1984: 37):

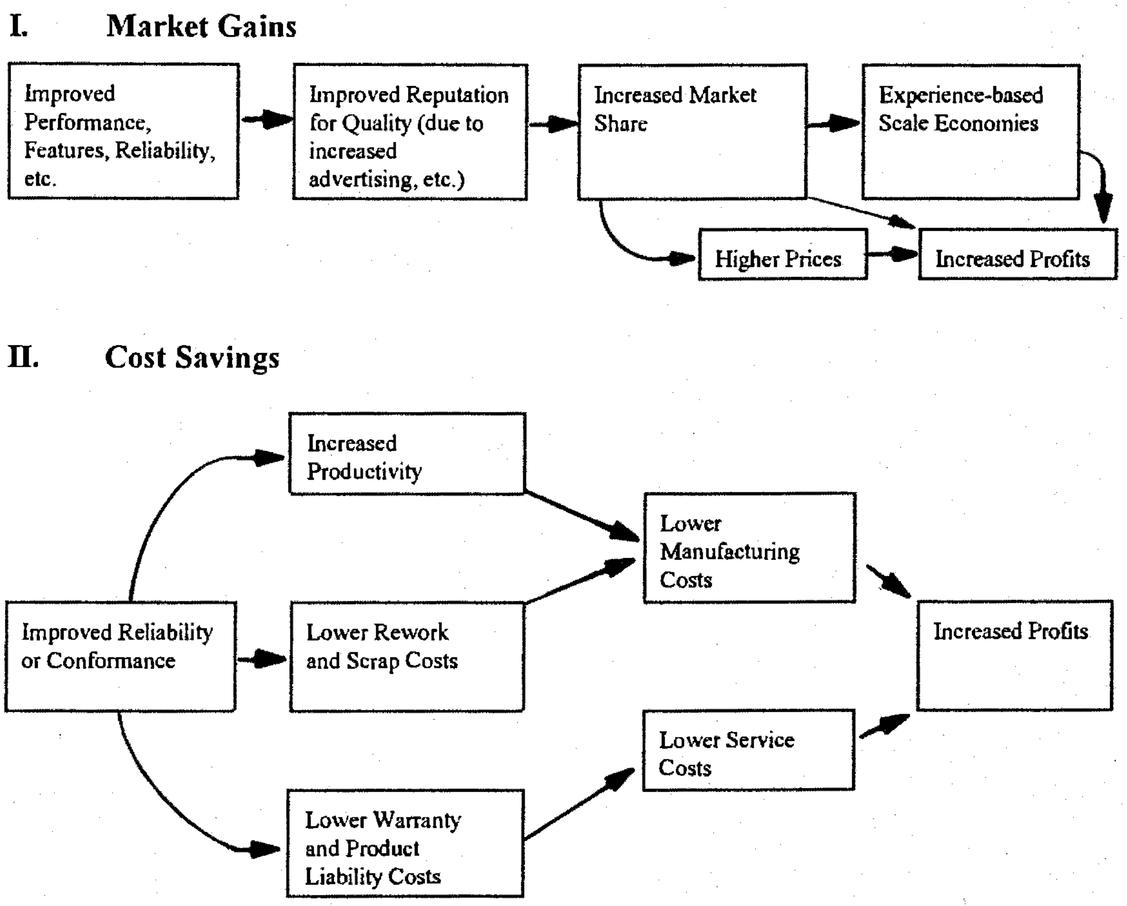

I. Market Gains
Improved Performance, Features, Reliability, etc.
Improved Reputation for Quality (due to increased advertising, etc.)
Increased Market Share
Experience-based Scale Economies
Higher Prices
Increased Profits
II. Cost Savings
Increased Productivity
Improved Reliability or Conformance
Lower Rework and Scrap Costs
Lower Warranty and Product Liability Costs
Lower Manufacturing Costs
Lower Service Costs
Increased Profits

Figure 2.1 Quality and Profitability

The quality dimensions can assist in determining the extent to which increased profits are achieved as a result of high overall quality, by segmenting the different aspects (Garvin 1984).

The ultimate aim of quality improvement programs is increased profits. Strategic planning and measurement of key aspects is thus of extreme importance in order to achieve the goal of improved quality and increased profits. These measurement aspects are discussed in chapter 3.

2.5.4 CONCLUSION

The empirical research on quality has produced mixed results with regards to the relationship between quality and the business performance indicators: cost, productivity and profitability. It is complex and difficult to predict, thus more precise measures of product quality is required. Garvin (1984) states that it needs to be established which dimensions are primarily a reflection of manufacturing skills, and which reflect design and engineering expertise. Only then can effective strategies for competing on the basis of product or service quality be devised and executed.

2.6 REDEFINING THE QUALITY DIMENSIONS FOR PROCESSES

We are currently functioning in a process-oriented world. Quality is no exception. The ISO (International Organization for Standardization) 9000 series of standards (to be discussed in chapter 3.4.7) refers to process quality and not product quality. Furthermore, the recognition of the importance of TQM for business has grown. TQM emphasizes process quality. Synergy exists between product quality and process quality and the dimensional aspect can be redefined for processes.

Redefining the dimensions within the context of processes will now be discussed.

A **process** is defined as ***a series of actions or operations in making or manufacturing or achieving something*** (Oxford Universal Dictionary 1981, s.v. "process").

1. **Performance**

   Attributes need to be identified that characterise the performance (the primary functioning) of the process. Once they are established, metrics can be defined to measure these attributes. For example, in the process of processing cheques, speed and accuracy are indicators of performance. Metrics can be the number of cheques processed per hour (speed) and the number of cheques **correctly** processed per hour (accuracy). Processes of the same type can be compared according to the performance indicators.

2. **Features**

Processes can usually be uniquely defined in terms of their "features", i.e. those things that distinguish them from other processes and that are regarded as being of particular importance. The role of the feature dimension in a process will be determined by the degree of importance of a particular feature to the user of the process. For example, easy access to information regarding the performance of the process may be regarded as a very important feature of the process.

3. **Reliability**

The reliability of a process can be described as the probability of a process' "failing" to succeed within a specific period of time. Careful strategic planning is needed beforehand to determine the context of defining what will constitute a process as failed and the development of the appropriate criteria. For example, the registration process of students at a university can be classified as failed if the records cannot be processed accurately and on time.

4. **Conformance**

This will indicate the degree to which the process conforms to preestablished standards. Within the context of processes, standards may not yet exist. In-house metrics need to be defined to establish the minimum requirements to which the process must conform. Conformance of a process should not be confused with being equal to a quality process as improvement above the minimum requirements is usually possible. Once again, taking the registration process of students as an example, one aspect of conformance can be defined as the processing of a *minimum number* of student records per day.

5. **Durability**

It can be defined as a measure of the "life" of a process, i.e. how long this process is going to be used and how "far" it can endure to handle change until it will be replaced with another process. If changes are made to the process, the period to replacement is extended. Organisations tend to do modifications to processes rather than to replace it, as replacement can mean an additional outlay in manpower and capital. The cost-effectiveness of this exercise and it's relation to the delivery of a quality service or product in the short as well as long term must be investigated.

An example may be how long the student registration process can cope if student numbers rose dramatically over a short period of time before the registration process needs replacement.

It ties in with the dimensions of reliability (failure of the process) and destructibility.

6. **Serviceability**

The service of a process can, as in the case of a product, be defined as the speed, courtesy and competence of "repair" to any part of the process. Repair will usually entail modifications and/or maintainability of the process. This aspect will influence the credibility of the process from the user's perspective. The response to and speed of repair, when a computer system that handles the registration of students goes down (both in terms of the personnel involved as well as the information technology), are some of the indicators of the serviceability of the registration process.

7. **Aesthetics**

This will be a subjective measure of the "user-friendliness" of the process, i.e. the accessibility of the process as perceived by the company, their suppliers and their customers. In terms of the registration process, it can, for example, refer to how students have experienced the process in the past.

8. **Perceived quality**

This will be closely related to aesthetics and refers to perceptions of what "quality" the process is supposed to deliver. Effectivity in, say, handling of the registration process by personnel and the technology involved, will result in higher perceived quality by the student.

9. **Adaptability**

Within the context of processes, this dimension is closely related to reliability and durability and will indicate the extent to which the process can be adapted to meet new constraints influencing the process.

For example, can the registration process be easily adapted to handle a 50% increase in student numbers?

10. **Destructibility**

This dimension can be interpreted in three ways:

1) It can indicate the environment-friendliness of a process, e.g. are all the chemicals used in developing a certain substance harmless to the environment?
2) It can indicate the "probability" of a process being wiped out, i.e. the degree of easiness with which the process can be destroyed through information technology failure or environmental factors.
3) Failure of process: Degree of possibility of total failure of process.

11. **Availability**

The process to be used must be "available" to the company wanting to use it. This will include resources (manpower, material and capital) and will also refer to the timeliness of the process. In terms of the student's registration process, it refers to the readiness of the process to handle registration when needed.

2.7 CONCLUSION

Careful analysis is required in understanding, describing and quantifying quality. A holistic view of quality is required in order to understand the approaches to and dimensions of quality, their interaction and their impact on business performance. Furthermore, as changes in the global economy is a certain phenomena, quality becomes an evolutionary concept, changing with time.

Literature findings concerning quality have to be checked for definitions used and interpreted accordingly. Once again, by viewing quality multidimensionally, sources of disagreement will not prevail.

Redefining the product quality dimensions to that of process quality dimensions seems natural and will result in an even better quality end result. By viewing process quality dimensionally, all aspects can be clarified and dealt with in a cohesive manner. The process quality dimensions can also assist in the strategic planning of processes within an organisation.

3. MEASUREMENT AND TOTAL QUALITY MANAGEMENT

Alice thought she had never seen such a curious croquet-ground in her life: it was all ridges and furrows: the croquet balls were live hedgehogs, and the mallets live flamingoes, and the soldiers had to double themselves up and stand on their hands and feet, to make the arches.

The chief difficulty Alice found at first was in managing her flamingo: she succeeded in getting its body tucked away, comfortably enough, under her arm, with its legs hanging down, but generally, just as she had got its neck nicely straightened out, and was going to give the hedgehog a blow with its head, it would twist itself round and look up in her face, with such a puzzled expression that she could not help bursting out laughing; and, when she had got its head down, and was going to begin again, it was very provoking to find that the hedgehog had unrolled itself, and was in the act of crawling away: besides all this, there was generally a ridge or a furrow in the way whenever she wanted to send the hedgehog to, and, as the double-up soldiers were always getting up and walking off to other parts of the ground, Alice soon came to the conclusion that it was a very difficult game indeed.

Alice in Wonderland (Lewis Carroll 1865)

The croquet game that Alice had to play very much depicts the ever-changing face of businesses today.

The aim of this chapter is to obtain a perspective on the multitude of measurement issues within the total quality management movement. The chapter is divided into four sections: background on the quality field, the strategic aspects of measurement in perspective, the measurement of quality management and quantitative structures for process improvement.

In the first section, an overview is given of the development of the total quality field, i.e. the period of inspection, then quality control, quality assurance and currently strategic quality management.

The second section consists of a comprehensive and integrated discussion on internal and external measurement aspects that relate to quality in it's strategic importance context. Aspects that are covered are the instrumental role of measurement in the link of quality to strategic and financial management; customer measurement; performance, measurement and quality; supplier measurement; quality and measurement systems; and the role of measurement in the Malcolm Baldrige National Quality Award.

The third section covers the measurement of quality management. An instrument for measuring the critical factors of quality management, developed by Saraph et al. (1989), as well as the use of this instrument to test the effect of organisational context on quality management by means of an empirical study (Benson et al. 1991), is described. A framework for quality management research and an associated measurement instrument (Flynn et al. 1994) are also described and compared to the work of Saraph et al. (1989).

The fourth section describes quantitative structures for process improvement currently used in industry as a vehicle to support, control and measure improvement.

The quality management maturity grid, cleanroom software engineering, software factories, quality function deployment, the seven planning tools and benchmarking are described in terms of what each constitutes and where it has been applied.

3.1 BACKGROUND

The development in the quality field, from the initial period of inspection to the current period of strategic quality management, is summarised. Garvin (1988) organises the discoveries in the quality field into four distinct "quality era's": inspection, statistical quality control, quality assurance and strategic quality management. Measurement has been, and still is, an integral part of the process of achieving quality. It is the vital link in the quality chain.

The summary is extracted from Garvin (1988).

3.1.1 THE PERIOD OF INSPECTION

The evolvement of mass production and the need for interchangeable parts were the reasons that necessitated formal inspection.

The key breakthrough (quality control wise) was the development of a rational jig, fixture and gauging system in the early 1800's. *Jigs and fixtures are devices that position tools or hold parts while they are being worked on, keeping them fixed to the equipment so that machining operations can be performed accurately and precisely* (Garvin 1988: 4). A system of gauges (gauges, like jig and fixtures, were based on a standard model of the product to ensure uniformity) was often used for ensuring accurate inspection of products.

Frederick W. Taylor (early 1900's) gave the activity of inspection added legitimacy by singling it out as an assigned task for one of the eight functional bosses (foremen) required for effective shop management: *The inspector is responsible for the quality of the work, and both the workmen and the speed bosses (who see that the proper cutting tools are used, that the work is properly driven, and that the cuts are started in the right part of the piece) must see that the work is finished to suit him. This man can, of course, do his work best if he is a master of the art of finishing work both well and quickly* (Garvin 1988: 5).

In 1922, inspection activities were linked more formally with quality control with the publication of G.S. Radford's "The control of quality in manufacturing". Although the primary focus was on inspection, emphasising conformance and its link with inspection, quality was, for the first time, viewed as a management responsibility and as an independent function. A number of principles that are regarded as central to modern-day quality control was also touched on: *the need to get designers involved early in quality activities, the need for close coordination among the various departments affecting quality and the association of quality improvement with increased output and lower costs.*

Quality control activities, at that stage, included inspection, counting, grading and repair.

Research conducted at Bell Telephone Laboratories proved to be the instrument for change leading to the following "era": that of statistical quality control, which will now be described.

3.1.2 STATISTICAL QUALITY CONTROL

In a memo dated May 1924, Walter A. Shewart proposed the control chart for the analysis of inspection data. This marks the beginning of modern methods of quality and reliability.

Shewart published his "Economic Control of Quality of Manufactured Product" in 1931. It gave the discipline of quality a scientific foundation. Garvin (1988: 6) remarks: *Much of modern-day quality control can be traced to that single volume. Shewart gave a precise and measurable definition of manufacturing control, developed powerful techniques for monitoring and evaluating day-to-day production, and suggested a variety of ways of improving quality.*

Shewart was part of a research group on quality problems at Bell Telephone Laboratories. The group also included Harold Dodge, Harry Romig, G.D.Edwards and later Joseph Juran. They were largely responsible for creating the discipline of statistical quality control as it is known today.

The critical aspects of process control and sampling within quality control, as well as the impact of World War II on the discipline of quality control, are briefly described.

3.1.2.1 Process control

Shewart was the first person to recognise that variability was a fact of industrial life and that it can be explained by using the principles of probability and statistics.

The entire analysis of process control grew out of Shewart's concept of statistical control: *A phenomenon will be said to be controlled when, through the use of past experience, we can predict, at least within limits, how the phenomenon may be expected to vary in the future. Here it is understood that prediction means that we can state, at least approximately, the probability that the observed phenomenon will fall within the given limits* (Garvin 1988: 7).

The process control chart, still one of the most powerful tools for quality personnel today, was also developed by Shewart.

3.1.2.2 Sampling

The second critical element in the growth of statistical quality control, sampling, was advanced by Harold Dodge and Harry Romig. An important development was the "Average Outgoing Quality Limit". It indicated the maximum percentage of defective units that a process would produce under two conditions: sampling inspection by lots, and the individual separation of good from bad items in all lots that had already been rejected on the basis of sampling.

Most of the original work was published in technical journals with limited circulation. The techniques were thus mainly used within the Bell companies.

3.1.2.3 Impact of World War II

The discipline of quality control grew tremendously in this time. Several aspects that indicate the growth of the discipline during this time include: the establishment of a committee in December 1940 to draft standards in the area of quality by the War Department, the publishing of these standards in 1941 and 1942 and the consequent establishment of a Quality Control section in the War Department, staffed to a great extent by statisticians from the Bell Laboratories.

Applications of the techniques were very successful. Training programs were initiated with the aim to extend the use of the techniques to other branches of industry.

Local societies for Quality Control were formed by former students of courses. The American Society for Quality Control (ASQC) was formed in 1946. The first United States journal on quality, called **Industrial Quality Control**, was published in 1944. This has later become Quality Progress, the official magazine of the ASQC.

By the late 1940's, quality control was established as a recognised discipline. The methods were primarily statistical, and the impact confined to the factory floor. This only changed when several key works were published in the 1950's and the 1960's that led to the era of quality assurance.

3.1.3 QUALITY ASSURANCE

In the period of quality assurance, quality evolved from a manufacturing discipline to one with broader implication for management. The tools for the profession expanded far beyond statistics. Four separate elements were involved in the evolution process: quantifying the cost of quality, total quality control, reliability engineering and zero defects. Together, they have led to a proactive approach to quality. Each of these will be briefly described.

3.1.3.1 The cost of quality

With the growing awareness of quality, a critical question arises concerning costs: How much quality is enough?

Joseph Juran tackled the question in the first edition of his **Quality Control Handbook** (1951). The famous analogy of failure costs to "gold in the mine" was proposed in the initial chapter of his book. This book became the profession's main reference at the time. Managers had a way to decide how much money to invest in quality improvement. It also underlined the importance of another principle, namely that decisions made early in the production chain had implications for the level of quality costs incurred later on.

3.1.3.2 Total quality control

Armand Feigenbaum proposed the concept of "Total Quality Control" in 1956: *The underlying principle of this total quality view...is that, to provide genuine effectiveness, control must start with the design of the product and end only when the product has been placed in the hands of a customer that remains satisfied....the first principle to recognize is that quality is everybody's job* (Garvin 1988: 13).

The existence of interfunctional teams became essential to make the system of total quality control work. Top management was ultimately responsible for quality. Feigenbaum, like Juran, also proposed careful measurement and reporting of the costs of quality.

Both Feigenbaum and Juran also indicated that a new function, quality control engineering, was necessary. This function would be involved in high-level quality planning, coordinating the activities of other departments, setting quality standards, and providing quality measurement.

3.1.3.3 Reliability engineering

Reliability engineering emerged in the 1950's. The objective was the assurance of acceptable product performance over time. It came about as a consequence of poor reliability of military components and systems.

The first step was to define reliability more precisely. Reliability was consequently defined as ***the probability of a product's performing a specified function without failure, for a given period of time, under specified conditions*** (Garvin 1988: 15). This definition, together with modern probability theory, led to formal methods for predicting equipment performance over time.

Prediction was only a first step. The discipline's goal was to improve reliability and reduce failure rates over time. Several different techniques were employed, e.g. failure mode and effect analysis (FMEA).

Furthermore, an effective reliability program required close monitoring of field failures. This reporting normally involved comprehensive systems of data collection as well as efforts to ensure that failed parts were returned to the laboratory for further testing and analysis.

Reliability engineering emphasizes engineering skills and attention to quality throughout the design process.

3.1.3.4 Zero defects

The concept of **Zero Defects** had its beginning at the Martin Company in 1961-1962. They delivered a Pershing missile to Cape Canaveral on December 12, 1961 with zero discrepancies. Another perfect Pershing missile was delivered on time, and was fully operational in less than twenty-four hours (the norm was ninety days or more).

Management concludes that the project's success was primarily a reflection of management's own changed attitude. The lack of perfection happened previously simply because perfection has not been expected. Furthermore, lack of attention as one of the main causes for worker errors has previously not been addressed sufficiently.

The company then designed a program with the goal to *promote a constant, conscious desire to do a job (any job) right the first time* (Garvin 1988: 17). The resulting program was called **Zero Defects**.

Garvin (1988: 17) summarises: *Martin's contribution thus lies primarily in articulating a philosophy -* ***the only acceptable quality standard was zero defects*** *- and in showing how it can be instilled in the workforce through training, special events, the posting of quality results, goal-setting and personal feedback.*

The Martin company's program was a major achievement. Quality control history at that time advocated that some non-zero level of quality was good enough. Crosby's (1979) (who worked at Martin in the 1960s) claim: *"that perfect quality is both technically possible and economically desirable" has rekindled many of the old arguments on how much quality is enough* (Garvin 1988: 18).

The debate around the zero defect principle still continues today.

3.1.4 EVOLUTION FROM INSPECTION TO QUALITY ASSURANCE

The following table extracted from Garvin (1988: 19) summarises the principal identifying characteristics for each period.

IDENTIFYING CHARACTERISTICS/ PERIOD	INSPECTION	STATISTICAL QUALITY CONTROL	QUALITY ASSURANCE
Primary concern	Detection	Control	Coordination
View of quality	A problem to be solved	A problem to be solved	A problem to be solved, but one that is attacked proactively
Emphasis	Product uniformity	Product uniformity with reduced inspection	The entire production chain from design to market, the contribution of all functional groups, especially designers, to preventing quality failures
Methods	Gauging and measurement	Statistical tools and techniques	Programs and systems
Role of quality professionals	Inspection, sorting, counting and grading	Troubleshooting and the application of statistical methods.	Quality measurement, quality planning and program design
Who has responsibility for quality	Inspection department	Manufacturing and engineering department	All departments, although top management is only peripherally involved in designing, planning, and executing quality philosophies
Orientation and approach	"inspects in" quality	"controls in" quality	"builds in" quality

Table 3.1 From Inspection to Quality Assurance

3.1.5 STRATEGIC QUALITY MANAGEMENT

Despite changes, approaches to quality remained largely defensive throughout the period of quality assurance. The main objective of the quality department was still the prevention of defects. Although a pro-active approach was pursued, quality was still viewed negatively. This view finally changed in the 1970's and 1980's when the strategic aspects of quality were recognised and embraced.

Quality is now starting to be linked to profitability, defined from a customer point of view and included in the strategic planning process. Quality is beginning to be regarded as a competitive weapon. These aspects will be discussed in 3.2.

3.1.6 SUMMARY

The development of the quality field is aptly summarised in the following quote: *This (the control chart) led to a broadening of the concept of inspection from emphasis on detection and correction of defective material to control of quality through analysis and inspection. Subsequent concern for product performance in the hands of the user stimulated development of the systems and techniques of reliability. Emphasis on the customer as the ultimate judge of quality serves as the catalyst to bring about the integration of the methodology of quality with that of reliability. Thus, the innovations that came out of the control chart spawned a philosophy of control of quality and reliability that has come to include not only the methodology of the statistical sciences and engineering, but also the use of appropriate management methods together with various motivational procedures in a concerted effort dedicated to quality improvement.* (Bossert 1991: v).

3.2 STRATEGIC MEASUREMENT ASPECTS IN PERSPECTIVE

The time has come, the walrus said
to speak of many things
of ships and shoes
of ***quality measurement*** *and......*

with apology to Lewis Carroll (1872)

3.2.1 INTRODUCTION

One of the cornerstones of TQM is the requirement for continuous and accurate measurement for every process that exists within the organisation, i.e. an internal view (Barrier 1992). One of the main weaknesses in implementing TQM programs is the failure to recognise the need to make these measurements (Stanleigh 1992). Data and measurable results are the bedrocks of TQM (Carpenter 1991). Problems need to be measured in order to be able to determine if the solution has brought any measurable gains. The sheer amount of information needed to trace quality problems in a complex organisational setting is still a constraint (Leonard & Sasser 1982). It is, however, in the pursuit of quality, important to see problems as opportunities. *An organization must put in place the systems, practices, culture, and rewards that will encourage people to be enterprising - to solve problems and to see and take advantage of opportunities* (Kanter 1987: 46).

There is also worldwide recognition that the impact and effectiveness of Quality Programs need to be measured, i.e. an external evaluation view. This view relates to the fact that the quality movement must advance to pragmatic, focused action (Miller 1992). Management-by-fact today infers that performance measurements are in place for all key processes of a business as well as for product quality as perceived by customers (Horst 1992). The emphasis should shift from the importance of quality to quality improvement.

People involved are usually unsure about what or precisely how to measure (Monoky 1992; Stanleigh 1992). It is a mistake to see measurement as an end in itself. A company doesn't earn money by making measurements. The trick is to avoid measurement of things that are irrelevant. Furthermore, it is sometimes possible to live with only approximate measurements of exactly the right things. This aspect is stressed by Kanter (1987) who says that by measuring everything as often as possible, all behaviour will revolve around the measures. Harari (1993) declares that one of the main reasons why TQM fails is the focus of TQM on internal processes rather than on external results. According to Harari (1993), preoccupation with internal performance measurements, conformance indices and technical specifications diminishes managers' attention to external factors like the constant shifting of customers perceptions and preferences, marketplace choices, technological advances and the possible product and service enhancements they could respond to. This can lead to a product or service that is outdated, too conventional, insufficient or irrelevant. The ultimate goal of quality is to add value to end-users.

Another area of concern is that TQM focuses on minimum quality standards. According to Harari

(1993), attaining minimum standards means that you will be able to do business, but is not a guarantee of success. Minimum standards do not define quality. The notion of equating quality with minimum standards is still a traditional viewpoint and not part of the multidimensional outlook on quality. Companies need to go beyond minimum standards. A point in case is the **Statistical Processes for Excellence in Quality Service** approach established by the Traveller Cheque Group (TCG) (Welch 1992) that will be discussed in 3.2.4.2 and 3.2.7.

The purpose of gathering information for quality improvement is to set corporate-wide benchmarks and standards that will place an organisation in a strong position to intervene before a problem occurs. The key is to determine what pieces of information is "critical to know". An appropriate measurement system needs to be developed. Activities should not be confused with results and the building of an infrastructure for quality (Benson 1992). Adequate definition, planning, implementation and evaluation is extremely important. Accountability through measurement is of utmost importance. Quality and Measurement Systems will be discussed in 3.2.7.

Internal process measurement and external customer measurement, together with internal workforce participation has been identified as the three common denominators that typify a successful TQM effort (Jordan 1992). Jordan stresses that the critical aspects, that is, the bottom line, the perceived quality of products and services, and the level at which the workforce produces have to be considered at the outset of any management initiative.

The following internal and external aspects of the strategic measurement of quality, i.e. the linking of quality to strategic and financial management, customer assessment, performance measurement and supplier measurement will be addressed and integrated. It will be discussed from a broader perspective, namely looking at it from a strategic multidimensional business viewpoint.

Monitoring the effectiveness of TQM efforts will be discussed in

1) the role of measurement in the Malcolm Baldrige National Quality Award (3.2.8) and
2) the measurement of quality management (3.3).

It is the author's perception that only by integrating and linking key internal and external measurements to quality, businesses will achieve optimum benefits. The use of this information by the organisation in its pursuit of quality, needs to be part of the strategic planning process.

3.2.2 DESCRIPTIONS

3.2.2.1 Internally focused measurements

Internally focused measurements, obtained through statistical process control (SPC) and other quantitative process improvement methods, *are used by the organisation to evaluate work process quality, output variation, and service quality performance improvement* (Jordan 1992: 47).

3.2.2.2 Externally focused measurements

Externally focused measurements are used to *quantify customer feedback on expectations/ satisfaction with service and product quality* (Jordan 1992: 47).

Another type of externally focused measurement is benchmarking.

Benchmarking is defined as ***the continuous process of measuring products, services, and practices against the company's toughest competitors and against companies regarded as industry leaders*** (Fenwick 1991: 65). As such, it is externally focused.

Benchmarking of processes within the organisation is now also taking place, i.e. it is used as an internally focused measurement. Benchmarking will be discussed in 3.4.5.

3.2.3 THE INSTRUMENTAL ROLE OF MEASUREMENT IN THE LINKING OF QUALITY TO STRATEGIC AND FINANCIAL MANAGEMENT

In a TQM environment, the shift from an inspection-oriented manufacturing-focused approach towards a defect-prevention and company-focused strategy is in place. Quality is considered to be an organisational goal and not just a functional responsibility (Leonard & Sasser 1982).

Quality considerations need to figure centrally in strategic planning (trade-offs, risks, performance and evaluation, and reward systems) and should be included as an integral part of all corporate review processes (Leonard & Sasser 1982). According to Leonard and Sasser (1982: 170), *the proper size of the quality function, its place in the organization, the breadth of its mission, and the nature of its role in the strategic process are all issues that need to be confronted in an organisation that aims for quality improvement.*

Davis (1992), in his conference report on the Fifth Annual Total Quality Conference presented by the Unified Technologies Center of Cleveland, USA in 1992 summarizes Juran, Crosby and Schonberger's (three acknowledged writers on quality) outlook on quality and the link to financial and strategic management as follows:

Juran emphasizes that top management involvement and planning is vital for quality improvement. He stresses the importance of the participation of senior management in measuring the influence of quality improvement on financial performance.

Crosby also emphasizes the need for top management involvement and a tighter linkage with financial performance. According to Crosby, one of the reasons why TQM does not become part of a corporate culture is because people don't measure its impact correctly. To quote Crosby: *Finance is what drive a corporation. Line and staff people need to measure the impact of quality in financial terms, otherwise top management, accountants, and finance people won't listen* (Davis 1992: 37).

Schonberger recommends the transferring of techniques that have been successfully used in manufacturing to administrative support and service jobs, e.g. the adoption of systematic data collection and Statistical Process Control as well as the use of visual management techniques and the elimination of unnecessary reporting. He advocates eliminating all cost accounting and variance reporting and suggests a yearly activity-based costing (ABC) audit in which the cost of all activities could be calculated for budgeting and resource allocation. Activity-based costing is also advocated by other researchers in this field such as Andreou (1991). Schonberger also stresses the need for all workers to document, control, and display their own processes. The activities that control the consumption of costs will then automatically be under control, and extensive cost reporting become redundant.

These viewpoints clearly demonstrate the important role of measurement in linking quality to financial and strategic management.

Davis (1992) stresses that TQM is not only quality improvement, it is also concerned with innovation, adding value, cost containment and productivity improvement. According to Harari (1993), it is the market-driven entrepreneurship and innovation that increases market value, not an obsession with doing it right the first time. Quality is more than correct processes. Measurement of quality has to take these added dimensions into account, i.e. strategic and financial aspects.

TQM in its widest scope, and strategic management are currently so interwoven that they have become undistinguishable. Achieving an integration of quality, strategy, and financial management is critical to the future of TQM. To succeed in the long run, quality management must be integrated with the strategic management process and blended into the customary market analysis, capital budgeting, and financial planning (Davis 1992). Harari (1993: 35) argues: *if quality truly is the centrepiece of doing business, it becomes everyone's responsibility and the cornerstone of strategy and operations, including budgeting.*

Fenwick (1991) defines benchmark criteria, strategic business objectives and key processes (defined as those that are determined to best satisfy the benchmarking criteria that one sets) as the three-legged strategy upon which the success of TQM rests. Fenwick (1991) advises that a model should be established to determine which processes need to be improved first in a business and how success will be measured.

In a study quoted by Fenwick (1991), The FORTUNE 500 companies in the United States were surveyed. Corporate executives were asked whether their companies measured a series of thirteen quality indicators identified in a previous study of Deming Application Prize winners. The Deming Prize was established in Japan by the Union of Japanese Scientists and Engineers (JUSE) in 1951 (Nakhai & Neves 1994). The 13 indicators (Fenwick 1991: 65) are:

Does your company track and report:

1) *The number of quality improvement projects completed?*

2) *Management attendance at quality councils?*

3) *Number of quality improvement projects linked to strategic goals?*

4) *Number of quality-related standard operating procedures?*

5) *Percentage of employees on Quality Improvement Teams?*

6) *Number of Quality Goals mutually established by managers/employees?*

7) *Number of formal quality service agreements established with customers?*

8) *Number of internal customer-supplier agreements?*

9) *Percentage of quality-improvement projects initiated at suggestion of customers?*

10) *Percentage of quality solutions applying to multiple departments/functions?*

11) *Hours of quality improvement training per employee?*

12) *Number of quality improvement teams with members from more than one department?*

13) *Customer complaints?*

The thirteen indicators are considered essential by executives in comparable Japanese corporations. It was found that the typical United States corporation tracks and reports on average on six.

Crosby (1992) emphasizes an equal concentration by executives on finance, relationships and quality. He again stresses the fact that management measures everything it cares about in financial terms. According to Crosby (1992), no company has placed the price of nonconformance into its accounting system and reports on it during management meetings. He argued that firms that deal with "acceptable levels" of nonconformance deals with the lifeblood of their organization: money and credibility. In doing things over, a lot of revenue is wasted. By not doing what they said they will, they are not going to satisfy customers. Witzke, quoted in Barrier (1992: 28) says: *When customers are happy, products are defect-free, deliveries are on time - all of a sudden you have got 30% more staff than you thought you had - because employees are spending less time correcting problems.*

This aspect is also stressed by Brown (1989) who indicates that very few organisations track the cost of non-conformance to quality specifications as part of their accounting statistics.

Kanter (1987) mentions the attention that needs to be paid not only to the visible mistakes, but also to the invisible mistakes. She cites one of Westinghouse's statements in its quality principles: *that an important source of waste is the failure to exploit a technological opportunity or use a new tool or technique* (Kanter 1987: 46).

Quality measures thus need to be evaluated jointly with financial measures, and the relationship between the two studied carefully. It should not be in conflict with each other.

Root cause analysis (Stanleigh 1992) has proved to be very successful in solving quality-relatedproblems within an organisation. By determining the root cause and measuring the impact that a problem has, "drastic" solutions to problems that management does not understand, can be avoided. This technique is also mentioned by Barrier (1992: 28) who quote Freese saying: *Take time to analyze the situation, do some statistical analysis if it's appropriate, get everybody you need together, and solve the problem forever. Inoculate your process.* Leonard and Sasser (1982) call this the identification of quality levers - that is, the exact location, cause, and pattern of distribution of each problem and the best way to resolve it. They add that the real challenge to management is to discover investments that will yield higher quality at lower unit cost.

Quality-related costs are much larger than currently shown in accounting reports (Stanleigh 1992). Costs can be anywhere from 20% to 40% of sales. These are usually included in the cost of ensuring "quality standards", but are avoidable (Stanleigh 1992).

Companies trying to implement TQM need to focus their resources on projects with a high potential for success rather than to try it on a company-wide basis. They need to pick areas of strategic importance and build on a foundation of measurable results. As long as TQM is integrated with the budgeting, strategy and performance-measurement process, it will not be treated as a temporary program (Davis 1992).

This agrees with the International Quality Study (IQS) findings discussed by Benson (1992) which asserts that TQM is a management system that must be designed and installed based entirely on the unique challenges that a company faces. The challenges must not be determined by what the company hopes to achieve but based on its current performance position.

The analytical structure of the study was designed to show which practices within 92 different assessment areas have an impact on the following three criteria: profitability (return on assets), quality (achieved quality as perceived by the end user) and productivity (value added per employee). Study participants were separated into three strata: low, medium and high performers, based on their current positions. Structural modelling was used. The results provide an indication for a company, given their strata from their profitability (low, medium, high) as a possible point of reference, the type of quality practises that they should follow and which to delay, to stay at least where they are or to get better.

It is important for companies not to confuse activities with results. According to Schaffer and Thomson (1992), activity-centered programs confuse ends with means and processes with outcomes. Companies believe that by carrying out the "right" improvement activities, actual performance improvements will materialise. Schaffer and Thomson (1992) refer to a 1991 study of more than 300 electronic companies, sponsored by the American Electronics Association, of which 63% out of the 73% that reported to have a total quality program under way failed to improve quality defects by even as much as 10%. They suggest "results-driven improvement processes that focus on achieving specific, measurable operational improvements within a few months" (Shaffer & Thomson 1992: 82). Only those innovations in management methods and business processes that can help to achieve specific goals are used. In a result-driven path specific targets are set and resources, tools and action plans are matched to requirements to reach the targets. Managers then know what they are trying to achieve, how and when it should be done, and how it can be evaluated. Shaffer and Thomson (1992) mention six reasons why activity-centered improvement programs fail:

1) It is *not Keyed to Specific Results.*
2) The scale of the program is too large and diffused.
3) *Results is a Four-Letter Word.*
4) *Delusional Measurement.* (Equating measures of activities with actual improvements in performance.)

5) *Staff- and Consultant-Driven.*

(Company-wide change programs installed by staff groups do not lead to successful transformation and activities suggested by consultants are rarely aimed at specific results).

6) *Bias to Orthodoxy, not Empiricism.*

There is no opportunity in activity-centered programs to learn useful lessons and apply them in future. It happens as a result of

- the lack of clear definition of beginnings and ends of activities and
- an inability to link cause and effect.

Four key benefits of a results-driven approach (Schaffer & Thomson 1992: 86) are:

1) *Companies introduce managerial and process innovations only as they are needed.*

Innovations were introduced incremental, in support of specific performance goals.

2) *Empirical testing reveals what works.*

The extent to which each approach yields results can be determined fairly quickly. Each improvement step is constantly assessed for contribution to meeting deadlines, so that performance improvement is an act of rational decision making based on evidence.

3) *Frequent reinforcement energizes the improvement process.*

There is no motivator more powerful than frequent successes. By replacing large-scale improvement objectives with short-term, incremental projects that yield tangible results, managers and employees can enjoy the psychological fruits of success.

4) *Management create a continuous learning process by building on the lessons of previous phases in designing the next phase of the program.*

Four aspects of starting a result-driven program (Schaffer & Thomson 1992: 89) are:

1) Ask each business unit to set and achieve a few ambitious short-term performance goals.

2) Periodically review progress, capture the essential learning, and reformulate strategy.

3) Institutionalize the changes that work- and discard the rest.

4) Create the context and identify the crucial business challenges.

The inevitable role of measurement runs like a golden thread through the literature on the link between quality and strategic and financial management. Interaction and trade-offs can only be assessed if they are measured. The secret of success lies in the correct and common sense application of the tool of measurement.

The anecdote "You can't control what you can't measure", today applies to every single aspect of business.

3.2.4 CUSTOMER MEASUREMENT

Customers.... are as hard to predict, anticipate, and understand as hyperactive three-year-old children on a diet of chocolate bars and sugar snacks (Schrock & Lefevre 1988: 236)

3.2.4.1 Introduction

An intense focus on customer satisfaction or the next step "customer delight" is an essential ingredient of any Quality Program. Businesses need to be customer-driven. The definition of customers includes external as well as internal customers (employees).

Horst (1992) regards the recognition that customer satisfaction equates to perpetuation of a business enterprise as one of the keys to successful TQM. Wellins, in Kendrick (1993: 13), summarizes today's outlook: *Business is recognizing the customer as the driver of product and service quality. Focusing on the customer takes a far broader meaning than customer service or customer satisfaction. Their requirements are becoming the focus of long-range planning.*

3.2.4.2 A customer measurement perspective: studies across different types of industries

In an article on quality in the telecommunication industry in the United States of America, Stout (1993) found that continuous quality improvement is driven by customers. Competition has placed a new emphasis on *whatever it takes to delight a customer* (Stout 1993: 18). She interviewed five companies: Alcatel Networking Systems, AT & T, Northern Telecom, DSC Communications Corporation and MCI.

Their views, especially on customers, and their measurement of customer satisfaction/delight will now be discussed.

Alcatel Network Systems (ANS):

Alcatel Network Systems (ANS) of Richardson, TX, is a growing part of Alcatel Alsthom, one of the world's largest manufacturers of telecommunications equipment.

Their goal is to be *the preferred supplier of microwave and lightwave equipment to the Bell operating companies interexchange carriers, independent operating companies, private, cellular, and others* (Stout 1993: 19).

For ANS *total quality performance means understanding who the customer is, what his/her expectations are, and the ability to meet the expectations without error, on time every time* (Stout 1993: 19). Their quality focus has changed from product control to process control.

They use a set of metrics called the Customer Satisfaction Index (CSI) to measure the needs of the customer. (Metrics are discussed in chapter 4.) The results are used to make changes that lead to process improvement.

AT & T:

It is based in New York and is a large telecommunication provider in the United States of America. It is one of the most diversified telecommunication companies in the world. Its business units are clustered in four groups: Communications Services Group, Communications Products Group, Network Systems Group and NCR.

Robert E. Allen, chairman and CEO of AT & T remarks: *we redirected AT & T to focus the talents and energies of our people on delighting our customers and winning in the marketplace* (Stout 1993: 20). AT & T measures customer expectations by looking at *performance, reliability, competitive price, responsiveness, features, on-time delivery, service and correct billing* (Stout 1993: 21). By *tracking the product or service that customers expect and the process where in that* *expectation is satisfied, reduction of waste, rework, and continuous quality improvement can be tracked* (Stout 1993: 21).

The plan-do-check-act cycle, originated by Shewhart, is used to assess their business units and to identify areas for improvement, thus a means to assess their efforts.

Northern Telecom:

Northern Telecom Ltd. is a leading global supplier of digital telecommunications switching systems.

Northern Telecom have five indicators that track key areas of concern: customer satisfaction, employee satisfaction, market share, return on investment (ROI) and quality. They focus on their customers by using quality function deployment (to be discussed in 3.4.4) and customer surveys. Their five marketing operations (headquarters in Toronto, Canada and McClean, Georgia; STC PLC (United Kingdom); STC Submarine System and Motorola-Nortel Communications Co.) survey each of their customers once a year to measure customer satisfaction for all products. Customer report cards are also analysed. The company also analyses areas of customer dissatisfaction and does a root cause analysis, a technique mentioned earlier in 3.2.3.

DSC Communications Corporation:

It designs and produces digital switching, transmission, access, and private network system products for worldwide telecommunications.

Primary objectives for their first customer survey in 1990/1991 were:

1) *Define and compare customer perception of leaders in the telecommunications equipment industry*
2) *Identify attributes most important in selecting a preferred supplier*
3) *Identify factors affecting the customer/supplier relationship* (Stout 1993: 22).

They established how customers perceived DSC and what factors are the most critical to address. They formed a customer satisfaction quality management team. Customer satisfaction issues were addressed by using customer surveys and applying proven problem-solving techniques to identify root causes of problems and then implement solutions. Repeated customer surveys help them to have the focus retained on the key drivers of customer satisfaction.

MCI:

MCI is the second largest interexchange (long distance) provider in the USA.

MCI has four focus areas when measuring continuous quality improvement. They are: Quality Performance Assessment; Measurement Analysis; Quality Management; Process analysis and Productivity Analysis. The corporate quality staff are responsible to facilitate the quality activities of each department.

It is clear that leaders in the telecommunication industry measure continuous quality improvement through the eye of the customer. The importance of the establishment and use of key performance indicators is emphasized. This aspect will be discussed in chapter 4.5.

The customer is the focal point when we try to measure service quality.

Berry, Parasuraman & Zeithaml (1988: 37) observe in their study on service quality (covering mainly the financial sector): *Customers assess service quality by comparing what they want or expect to what they actually get or perceive they are getting. To earn a reputation of quality, an organisation must meet or exceed customer expectations.*

According to Berry et al. (1988: 37), customer expectations cover five areas:
Tangibles: *the physical facilities, equipment, appearance of personnel.*
Reliability: *the ability to perform the desired service dependably, accurately and consistently.*
Responsiveness: *the willingness to provide prompt service and help customers.*
Assurance: *employees' knowledge, courtesy, and ability to convey trust and confidence.*

***Empathy:** the provision of caring individualized attention to customers.*

The reliability dimension proved to be the most important aspect influencing customers, irrespective of the service area chosen. Berry et al. (1988) conclude that the most important aspect of service provision is that the service provider does exactly what they promised to do.

Berry et al. (1988) also point out the importance of the human element in services provision. Three of the five characteristics: responsiveness, assurance and empathy, result directly from human performance.

Once again, key areas for determining service quality was established by the researchers. These need to be addressed within the company and their processes changed to accommodate the expectations of customers. Only then will the knowledge gained from the customer be of optimum benefit to the company.

A quality measurement tool called the Service Tracking Report (STR) was developed during the period 1982-1983 by the American Express Company (AMEXCO) Traveller's Cheque Group (TCG) to establish the quality of their service from the viewpoint of the customer (Welch 1992). One of their three quality ordinals that serve as a strategic base for establishing their quality objectives is a commitment to *prevention-based work processes and data systems, with identifiable standards, targets, and continually improving results* (Welch 1992: 464). This ordinal thus encourages "management by facts".

Through the use of this measurement tool, TCG *began to face reality and take a hard look at facts* (Welch 1992: 465). According to Welch (1992), managers must be trained to develop analytical and problem-solving skills in order that they can follow the principle "Use the right facts, use the facts right".

TCG's three customer groups (sellers, purchasers and acceptors) were firstly surveyed and their answers were grouped into three categories of expectation: accuracy, timeliness and responsiveness.

The STR was designed to monitor accuracy, timeliness and responsiveness to customer needs. Twenty-eight key indicators, derived from customer input, were identified and measured daily and reported weekly in the STR. **Percent achievement** was used as the primary format for STR measurement for five years. Performance ratings and compensation of customer service executives were linked to the successful implementation and use of service tracking. The STR also helped to bring work processes under control and it facilitated improvement.

After five years (1987) major quality measurement changes were made by tightening of original standards by an average of 18% and a reevaluation of percent achievement as measure. An additional measure, namely **percent met standard** was decided upon to measure service quality. By using this measure, it was possible to track the portion of the employee population that actually met customers' quality standards. Using both **percent achievement** and **percent meet standard** it provided "a complete picture of how well and how persuasively TCG was giving its customers what they were looking for" (Welch 1992: 466). An accurate view could be obtained of process performance and improvement opportunities by analysing trends in the data. This underlined the importance and relevance of measurement in an organisation. In order to be able to keep track with the changing needs of customers a number of additional instruments are used.

In the 1990's, TCG had to address another problem: the percent met standard was no longer sensitive enough to expose areas of nonconformance. Results were above 99 percent compliance to standards. As the remaining 1% was still crucial, the **Statistical Processes for Excellence in Quality Service** approach was developed. This will be discussed under the heading of Quality and Measurement Systems in 3.2.7. TCG thus linked customer measurement, performance and process control successfully.

Thomas Interior Systems, designer and reseller of office furnishings, have also turned to the customer in order to be able to measure quality. From internal and external customer interviews they have established what they should measure (Barrier 1992).

The three Malcolm Baldrige award winners of 1991, to be discussed in 3.2.8, have all emphasized the important role of their customer satisfaction measurement systems.

3.2.4.3 Research on customer satisfaction measurement

Customer satisfaction has, in the past, almost always been measured negatively, that is, mainly in terms of complaints and service calls. A further matter of concern is the fact that customer complaints were found to be of "major or primary importance" in only 19% of banks, 26% of hospitals and 26% of computer manufacturers in America. In contrast, computer manufacturers in Germany and Japan (60% and 73% respectively) use customer complaints (Harari 1993).

Customers would rather switch suppliers than complain. According to a study of high-tech equipment buyers, noted in Gordon (1993), 63 % of all dissatisfied customers will never do business with that company again. Ninety percent (90%) of those dissatisfied customers will remain loyal to the supplier if the supplier resolves its problems.

Currently, customer satisfaction research is a required component of quality programs, which include ISO 9000 certification, Six Sigma, the Malcolm Baldrige Award and the Shingo Prize (Gordon 1993). It helps companies to improve business and to keep track of customer issues. Hyde (1991) states that customer satisfaction measurement is one of the best techniques to emerge from the quality management movement. By combining this with Statistical Process Control (SPC), performance and rework indices, and other measurement instruments an organization can direct an array of techniques to assess quality costs and process improvement (Hyde 1991).

Cravens et al. (1988) state that the central idea which underpins the concept of quality is that each part of the organisation has customers which it should seek to satisfy. All parts of an organisation should look systematically at the process by which they satisfy their own customers in the production chain from the acquisition of raw materials to delivery to the final customer and provision of after sales service.

Cravens et al. (1988) identify alternative approaches to measure quality. In their view, the most appropriate approach is based on measurement of the perception of customers of important product or service features. Customers are asked to rate the company against competitors on key performance dimensions which are important to them rather than the dimensions that the company considered important. The company must then identify internal processes which may influence these perceptions and seek to introduce performance measures which can be used to measure the effectiveness of these processes.

Linking up with the approach of Cravens et al. (1988), a systematic approach suggested by Salter (1991) for measuring customer satisfaction is summarized:

1. **Define goals and how information will be used**

   A common failure of customer satisfaction research is the lack of clear, comprehensive, measurable goals. ... key parts of a company must be involved in setting objectives for customer satisfaction measurement and management (Salter 1991: 9)

   It is also very important to determine how the information is going to be used. *Careful analysis of strategic and tactical organizational applications will ensure that issues of design, sample, analytics, reporting, and deployment are structured to provide customer-focused information that can be acted on most effectively* (Salter 1991: 9).

2. **Discover what is important to customers and employees**

   The attributes that form the perceptions and expectations of quality and satisfaction need to be identified in this phase. This information is gathered through qualitative techniques. The research will lead to a comprehensive list of important attributes. Techniques then need to be applied to eliminate redundant or related attributes and to agree on those that will be used for subsequent measurement as key drivers of satisfaction.

3. **Measure critical needs**

   Critical needs assessment is used to measure the relative importance of the attributes and the company's competitive performance on those attributes. Quantitative information is obtained and trade-off techniques, instead of importance scaling, provide improved discrimination pertaining to the relative importance of attributes. Information that should be obtained is *the relative importance of key drivers of satisfaction; competitive performance on these critical attributes; site-specific performance, depending on sample size; cross-market segments with specific service needs; value-adding performance relative to expectations and specific gaps between performance and importance*

(Salter 1991: 9).

4. Act on the information

Activities to improve customer satisfaction can now be planned by *operationally defining and functionally deploying customer requirements* (Salter 1991: 9).

Techniques such as Quality Function Deployment (QFD), Pareto Charts and Cause-and-effect diagrams can be used by teams to improve processes.

5. Measure performance over time

Salter (1991: 9) comments: *Periodic measurement of how a company and its competitors perform on the key drivers of satisfaction reveals the rate at which customer satisfaction is improving or declining. ... Frequency of measurement should be determined by market dynamics and allow for sufficient time for change to become measurable.*

Good customer surveys is a critical component in the measuring of customer satisfaction/delight. Cassell (1992: 65) suggests seven steps to a successful customer survey:

1) Plan the survey
2) Perform a self-assessment to meet customers' expectations
3) Organize backup documentation
4) Practise dry runs
5) Implement pre-survey activities
6) Launch the survey
7) Implement post-survey activities

Gordon (1993) describes a methodology that has been used in customer satisfaction programs to measure customer satisfaction in three electronic industries in the USA. Companies provide a confidential list of 10 customers whom they have served in the past year to market research companies. Market-research analysts then conduct a telephone interview with 7 of the 10 customers and rate the responses by using a 10-point scale. The customers are asked to give ratings and reasons for ratings in five categories of service. Gordon (1993: 41) continues his explanation: *Each participating supplier receives all ratings and reasons given by customers. Participants also receive the average, high and low ratings given to their industry as a whole, and recommendations for improving their customers' level of satisfaction. Participating suppliers are not identified by name except to their customers during the interviews.*

Invaluable benchmarking information can be obtained through such an exercise. The topic of benchmarking, which is part of the measurement process, will be discussed in 3.4.5.

Furthermore, delays in responding to internal customer (worker) requests, directly or indirectly, add up to a failure to meet external customer requirements. Benchmarking of processing your own work within an organization is necessary to be able to rectify poor internal customer service (Chaleff 1993). Internal benchmarking is also an excellent way of achieving incremental gains within a business unit or company (Benson 1992).

Recent promising methodologies include the work of Karnes et al. (1995) and Holcomb (1994). Karnes et al. (1995) incorporate the eight quality dimensions suggests by Garvin (1984) to measure quality from the consumer's perspective. The Analytical Hierarchy Process, a pairwise comparison approach, is used as the technique to measure overall quality. Holcomb (1994) suggests a methodology for customer service measurement through the utilization of the Taguchi strategy.

3.2.4.4 Conclusion

As customers are the lifeblood of any organisation, their perceptions of services and products are very important. These can only be obtained by means of qualitative and quantitative data. Measurement of customer satisfaction/delight is thus critical in obtaining the required information.

However, the recent International Quality Study (Benson 1992: 34) finds that *increased participation by customers does not demonstrate positive impact for companies at any performance level.* The level of customer research and measurement thus needs to be planned carefully within the context of the business so as to achieve the required results.

The ultimate benefit of customer measurement lies in improving quality throughout the company, meeting quality program requirements, creating loyal customers, and earning a reputation for caring about customers' perception of quality.

A further important aspect is the difference between measuring merely customer satisfaction as opposed to customer delight as well as the link of these to quality. This aspect has also been highlighted in chapter 2.4 and warrant further research.

3.2.5 PERFORMANCE, MEASUREMENT AND QUALITY

The field of performance measurement is beyond the scope of this dissertation. Only aspects related to quality will be briefly described.

Financial performance measurements are not yet adapted to the total quality management environment, an aspect that has also been discussed in 2.5 and 3.2.3. Allen (1991) mentions that previously, in the time of mass production, the focus was on average unit costs. Standard costs were directly linked to the budget and an "adverse" variance was a signal of inefficiency. Currently, it can also *signal that the product mix is more varied, and/or biased to more elaborate offerings* (Allen 1991: 19). If performance is measured by reference to budgeted average unit costs, it will, according to Allen (1991), motivates the production side to resist satisfying customers' needs!

Allen (1991) suggests that the answer to the above problem is acknowledgement that standard costing and budgetary control can be developed in different directions, e.g. the customization of standard costs for a specific company. He also stresses that today's business environment is more uncertain and that accountants *need to accept and work with margins of error: neither the accuracy nor the precision associated with traditional accounting are possible* (Allen 1991: 19). Furthermore, according to Allen (1991: 19): *tailoring of products and services has, by definition, meant that many key decisions are made in respect of a particular customer, or group of customers.* Customer profitability can then not be determined only by additional analysis of existing cost accounting data.

The supplier-customer relationship within an organisation is another point of concern. A buyer, should not be judged only by reference to the price of materials, but also in terms of availability, ease of handling and failure rate of the material supplied to the production functions.

The linking of compensation to performance measurement based on quality indicators is still a controversial issue. The IQS study (Benson 1992) found that the practise of linking quality efforts to compensation programs only fuel frustration when the reality is that the infrastructure of the organisation does not yet have the capacity to deliver the quality envisaged. Barrier (1992) also notes that profound disagreement exists among experts over how compensation should be determined under Quality Programs. Welch (1992), on the other hand, mentions that, at the Traveller Cheques Group (TCG), performance ratings and compensation of customer service executives were linked to the successful implementation and use of service tracking. At Motorola, Ford and Federal Express quality indices are also important determinants of management compensation. Harari (1993) mentions an Ernst & Young study which found that fewer than 20% of organizations in the auto, computer, banking and health care industries have quality performance measures that play a key role in determining senior management pay. Profitability still matters the most in all four industries.

A motivational aspect of recognition of performance is the publicity value that creates a culture of pride in which everybody feels they must live up to the level of achievement set by the people who were singled out as role models (Kanter 1987). According to Kanter (1987: 48): *challenge - opportunity - is one of the greatest untapped potential rewards that most organisations have. It doesn't cost anything to give people opportunities and yet it often pays off in problems solved and innovations developed.*

Establishing objective measures of performance for quality improvement within a Research and Development (R&D) group is difficult. Measurement systems that have been implemented to assess R&D productivity and innovation are subjective, and the establishment of reward and recognition systems based upon individual contributions to quality in R&D are absent in many organisations (Montana 1992). According to Montana (1992), effective measurement systems involving time, cost, efficiency, and customer focus need to be instituted and monitored by R&D management. He argues that these measurements will not only serve to accurately track success of the R&D quality process, but also to stimulate the interest of the technical staff in applying their expertise to quality matters. A list of R&D performance measurement criteria can be found in Montana (1992).

Performance measurement is equally important in manufacturing. Continuously measuring factory productivity and product or service quality as perceived by the customers is management based on fact. *How we measure performance strongly influences how we play the game .... Zero errors will not make a company competitive if they are not scoring runs. We must measure the efficiency and quality of output. Demonstrating (by measuring) performance excellence in the processes and product validates our TQM strategy and confirms our customer commitment* (Horst 1992: 46).

Performance analysis tools such as the productivity equation:

$$p = R \times A \times y$$

where R is the production rate

A is the process availability

y is process yield

and the Taguchi quality loss function are important tools that can be used by companies (Horst 1992).

The interaction between performance measurement and quality need to be carefully analysed within the context of the company. Shin, Riel & Sink (1988) summarizes: *A measurement system that is embedded in an overall performance management process must encompass bottom line considerations and include all other performance criteria involved in the success of a firm. Factors such as effectiveness, productivity, and quality must also be measured if the management process is to be successful.* Measurement systems are discussed in 3.2.7.

3.2.6 SUPPLIER MEASUREMENT

3.2.6.1 Introduction

The supplier is a crucial part of the partnership of producing goods and services (Yovovich 1991).

In the same way that loyal and satisfied customers are important, loyal and good suppliers are important (Yovovich 1991). Harari (1993: 36) mentions that companies such as Xerox and Ford are now bringing in a small group of selected suppliers as *long-term partners, giving them training, sharing data and cost savings, allowing them to access central databases via electronic data interchanges, and working collaboratively with them on common problems, new ideas and potential opportunities.*

This is also agreed upon by Barrier (1992: 23) who quote Noel Pooler (owner of Pooler Industries): *They (large firms) are attempting to reduce the number of suppliers that they have - they want long-term contracts, fewer and fewer suppliers and better and better quality.* He added that Pooler's customers look at the quality of every aspect of the company: *how it handles deliveries, how rapidly it responds to engineering changes, how quickly and politely its phones are answered.*

These aspects are part of the Just-in-Time (JIT) philosophy currently followed by many companies. JIT is shortly described as, in the broad sense, *an approach to achieving excellence in a manufacturing company based on the continuing elimination of waste (waste being considered as those things which do not add value to the product). In the narrow sense, Just-in-Time refers to the movement of material at the necessary place at the necessary time. The implication is that each operation is closely synchronized with the subsequent ones to make that possible* (Apics Dictionary 1987, s.v. “JIT”).

3.2.6.2 Supplier measurement approaches

Measurement and feedback is one of the most important steps in the supplier quality management cycle (Broeker 1989). Supplier measurement, according to Broeker (1989), should contain all critical variables such as quality, delivery and price. Quoting Broeker (1989: 68): *Price measurement should include the cost of non-conformance traceable to the product. The cost of an item should reflect the initial purchase price plus the added costs resulting from items such as scrap, rework delays, field failures and poor supplier quality. These costs might greatly exceed the savings achieved by buying from the lowest bidder.*

Purchasing managers often lacked information on quality-related costs. It is effective to develop customer measurements along with supplier measurements for major material purchases (Broeker 1989). The importance of the integration of customer and supplier measurement is thus once again emphasized.

The reverse market-research approach (to survey suppliers) followed by Motorola (Yovovich 1991) is also followed by other Baldrige Award-winning companies like Marlow and Solectron (Davis 1992). The Malcolm Baldrige Award will be discussed in 3.2.8. Questionnaires are an integral part of this process and are used to measure suppliers perceptions. An additional benefit is the attainment of good benchmarking information.

An area that needs investigating is an aspect mentioned by Mr. Stork of the Motorola company who was quoted in Yovovich (1991: 29): Suppliers' main quality problem can be that they have to many customers. *Because customers can have sharply differing needs, a supplier's efforts to meet the varied needs of all the different customers can cause the suppliers to make errors, and the intelligent solution to their total-quality program is to reduce their customer-base.* This aspect can be assessed by means of a correct measurement system. Quality and Measurement Systems are discussed in 3.2.7.

3.2.6.3 Supplier quality certification

Supplier quality certification is a means to determine the suppliers that can produce all the parts ordered defect free and deliver them just-in-time. It implies that the suppliers who obtain the certification have reached a certain level of excellence. Stout (1993) mentions the supplier certification of Alcatel Networking Systems designed to ensure the ability of suppliers to deliver quality components on time, every time.

One of the International Quality Study (IQS) outcomes was that performance gains occur across the board for companies that use formal supplier certification programs (Benson 1992).

Inman (1990) discusses quality certification of suppliers by Just-In-Time (JIT) manufacturers. The definition for quality certification is in the form of a set of requirements for the supplier (Inman

1990: 58):

1) *Statistical Process Control must be utilized.*
2) *They need to have a quality assurance plan (a set of written procedures).*
3) *They need to make delivery commitments.*
4) *They need to be part of a formal education program.*

These four requirements are considered as a basis for the certification process (Inman 1990). However, in a study conducted by Inman (1990), he found that Stowe's definition were not standard for all the JIT companies surveyed. Further research is required in the modelling of quality certification of suppliers.

3.2.6.4 Conclusion

Supplier measurement, together with customer measurement, provide a company with invaluable information. The correct use of this information is critical on determining the usefulness and benefit to the company. This aspect is discussed in the next section: Quality and Measurement Systems.

3.2.7 QUALITY AND MEASUREMENT SYSTEMS

A major problem is the temptation to stress the management philosophy aspects, to hawk the importance of quality without really stressing the means and methodologies that must drive a process premised on continual improvement

(Hyde (1991: 20) on TQM programs).

The incorporation of quality in measurement systems in still in its infancy.

Quality is measured primarily through the management accounting system and the operating control system (Andreou 1991). The management accounting system measures costs of rework, scrap, and warranties. The operating control system uses statistical measures and techniques. Quality indicators are mainly measured through the operating control system. The level of detail obtained is not enough for effective decision making (Andreou 1991). This view is also shared by Shin et al. (1988) who claim that the classical measurement system that has been and is still being used, is the traditional accounting system. The system only provides information on *efficiency, profitability and budgetability* of an organisation (Shin et al. 1988: 453). They include quality in their view of a complete measurement system.

Leonard and Sasser (1982) stress the shortcomings in the current measurement and performance systems which ignore quality areas. The way by which managers measure, estimate and account for quality-related issues needs reexamination. Measurement and estimation of quality decisions for the short and long term must be taken on a more formal basis. This is also mentioned in Brown (1989). He indicates that measurement of quality and its associated aspects need to be done outside the accounting function, e.g. the cost of non-conformance may be very difficult to calculate using existing measurement systems.

The role of measurement systems, in relation to quality, is discussed in Hyde (1991). He suggests that, for a Quality Program to work, it needs as first step, to have measurement systems in place, accessible to everybody in the organisation. Accessibility is also mentioned by Kanter (1987). She emphasizes the access of employees to the three key power tools in an organisation: information, support and resources. According to Kanter (1987), change master companies tend to make more information more available to more people at more levels through more devices. These devices include oral and written communication. The companies also emphasized timely information. Information is always needed wherever there is change. Hyde (1991) suggests, as minimum requirement, the following basic quality measurement systems: process improvement and statistical process control, group performance and rework indices, and customer and client feedback analysis. Keith (1994) mentions that data analysis tools need to be provided through the Management Information Services (MIS) group within an organisation that allow users access to key performance information.

The involvement of employees from the design stage in developing the measurement systems to ensure that the information and analysis generated has useful meaning is recommended by Hyde (1991) as the second step. This requires training in quality measurement and quality analysis for all employees, including managers.

Thirdly, Hyde (1991: 19) suggests that effort is required on *reforming work redesign, compensation, performance evaluation, and training and development systems to complement a quality management process.* Personnel, budgeting and resource systems need to be realigned to complement quality systems (Hyde 1991).

The entire business process thus need to be included when creating a Quality Information System (QIS) (Keith 1994). According to Keith (1994: 29): *QIS requires the systems department to develop, implement, and champion a methodology that looks at the business goals and develops activities that might or might not include a computer system to meet those goals.*

Garvin (1983) found that the best performing companies have excellent information systems where information regarding quality is on time, more accurate and complete. The timeliness of information has also been mentioned by Kanter (1987) in relation to companies that has mastered the ever-changing global business environment.

The success of an information system requires mastering of the details. Variation in the level of reporting detail correlates strongly with the quality performance of an organisation (Garvin 1983). Important differences between products may not be detected if data is highly aggregated. Design errors are also not detected early if precise reports are not available. Stout (1993) cites the example of the Alcatel Networking Systems (ANS) company who, by paying close attention to process detail and not just to the end product, have, in the end, delighted customers.

Another aspect is that information systems exist in organisations but are not used. Schlange [reported in Godfrey (1993)] studied quality information systems in six companies. He found that only one of the companies actually used the quality information - the Xerox company. *It closed the loop and turned the data collected into useful information and then turned the information into action. The information was used to improve the next generation of products, improve business processes, reduce cycle times, improve distribution, improve field service, better understand the needs of customers, and design products and services to meet those needs.*

An aspect of measurement within organisational context that is often overlooked is that the mere act of measuring human processes changes them. Measurement should be limited to those items that will really be used, bringing us back to the aspect of establishing the key factors that need to be measured. Measurements are expensive and disruptive and can degrade the processes we are trying to improve.

Data can thus be biased and distorted by the means used to acquire them (Hill 1992). This aspect is also stressed by Fechter (1993). Unfiltered information flow is critical to the success of any organisation.

Kaplan and Norton (1992) acknowledge the fact that the measurement system of an organisation affects the behaviour of managers and employees. They propose the "Balanced Scorecard" that consists of a set of measures to give managers a comprehensive view of the business. It includes a balanced representation of financial and operational measures. It consists of a financial -, customer - , internal business - and 'innovation and learning' perspective. The balanced scorecard represents a fundamental change from previous performance measurement assumptions. It puts strategy and vision of the company in the center and not control.

The availability of data to monitor quality effectively is one of the major stumbling blocks in setting up effective quality information systems. The absence of an infrastructure for the collection, organisation and processing of data is one of the major causes of this problem. Identification of areas where data and information should be collected within the organisation is of vital importance if the strategic importance of quality is to be accommodated within the financial framework (Andreou 1991).

The critical importance of an efficient and effective data collection process will be discussed in chapter 4.

Wood and Preece (1992) suggest a Measurement- based Approach to Quality (MAQ). They stress the fact that it is important to link a mathematical appreciation of the available techniques with a social scientific understanding of social processes, structures and working practises in organisations. The approach needs to be designed to work in the given context. A model of an MAQ design and adoption is given in figure 3.1 (Wood & Preece 1992: 43).

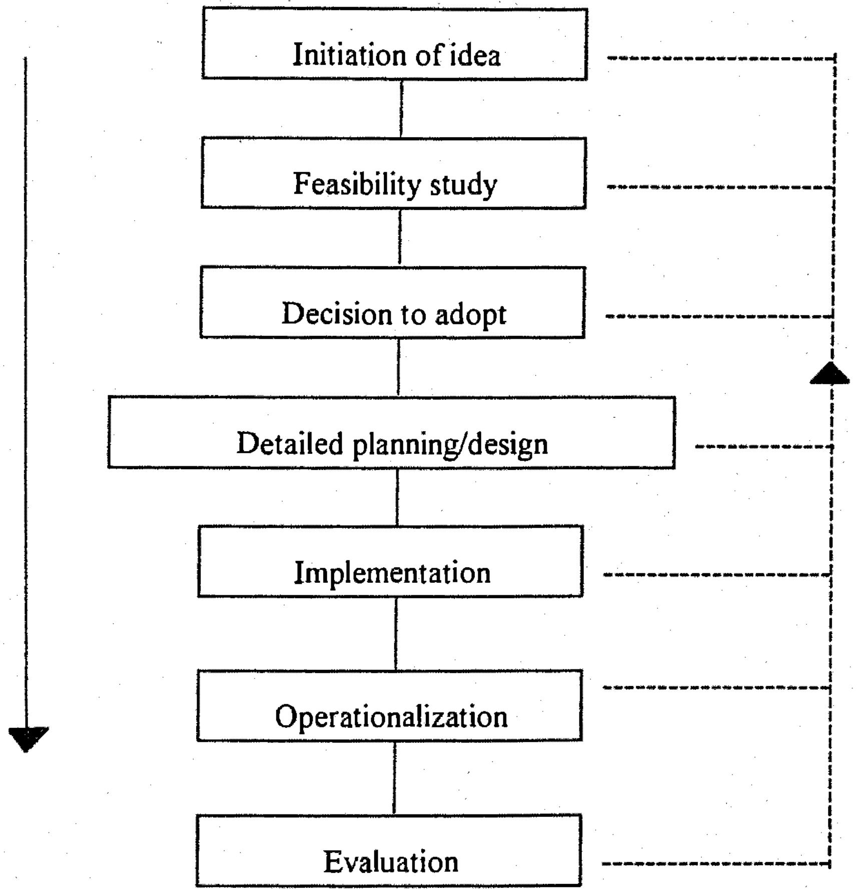


Figure 3.1 Model of MAQ Design and Adoption

Wood and Preece (1992), after studying three (two manufacturing and one software) companies on their use of quality measurement, draw the following conclusions:

1) The initial objectives for using an MAQ, e.g. cost savings or customer pressure may have implications for the detailed design and implementation.

2) Designing effective MAQ's may be more difficult than initially realized.
If a list of so-called "standard" procedures are selected, difficulties arise as no real situation is "standard".

3) *There may be a conflict of interest between the stated aims of the quality management*

system, the aims that must be met to satisfy customer pressures, or pressures from elsewhere in the organisation, and the perceived interest of the people implementing the system (Wood & Preece 1992: 51).

4) One easily overestimate people's understanding of technical terms and results. Techniques can be misunderstood and thus misapplied.

5) The importance and necessity of evaluation and thus feedback to improve the system is emphasized.

Wood and Preece (1992) make the following recommendations:

1) The objectives of the quality management system should be clearly specified before making any attempt to design an MAQ. A cost benefit analysis needs ideally to be included in this process.

2) *A proposed MAQ should be designed and evaluated as a whole system, incorporating a framework of mathematics, of skills, knowledge and experience needed by the users/implementers to operate the MAQ effectively, of the appropriate procedures and control, monitoring and reward structures and, possibly, computer hardware and software* (Wood & Preece 1992: 52). Restructuring can encounter resistance and strategies must be developed to deal with this.

3) Training programs usually teach techniques and how to do it. More important is a thorough understanding of the techniques in order to interpret the results and know how to act on them.

An excellent example of a measurement program that incorporates quality is the one that is mentioned in Welch (1992). The Traveller Cheque Group (TCG) developed a **Statistical Processes for Excellence in Quality Service** program. Their approach will now be described.

Service tasks are not quantifiable in the same way as manufactured products. Furthermore, two important differences exists: There are rarely any formal "specification limits" assigned to service tasks and secondly, the notion of "process capability" goes undefined in the service industry.

A core principle for the use of Statistical Process Control (SPC) in the service industry is the following: *Service industry improvement trends favour a standard. For measurements related to timeliness, the improvement trend should favour the lower specification limit, towards the zeroline; for measurements related to accuracy, the trend should favour the upper specification limit of 100 percent error-free delivery* (Welch 1992: 469).

According to Welch (1992) , one of the main problems in applying Statistical Process Control (SPC) to the service industry is the need to maintain a balance. At TCG, under or over delivery in any of their three customer categories (accuracy, timeliness and responsiveness) would negatively impacts the delivery of quality. The human side of the business remains extremely important.

Currently, the Six Sigma approach to quality improvement is adopted by TCG. It can be characterized as a statistical approach to quality improvement goal setting. Personnel are not directly involved in statistical analysis. A Quality Assurance and Engineering Group handles this aspect. The Service Tracking Report (STR) is used as the communication tool with employees.

Welch (1992: 471) concludes: *An organization's facts, statistics and quality indicators provide it with sight. ... by building values, measurement tools, and work processes that depend on both sight and foresight, organizations can have total quality systems that envision ways to continuously increase customer satisfaction and business profitability and help turn those visions into reality.*

CONCLUDING REMARKS

The literature thus all points towards an integrated system, where accounting, process control, customer and supplier measurement as well as performance criteria measurement, are included.

Quality measurement systems are navigational tools to get to the unlimited destination of quality improvement but need to be used with expertise (Hyde 1991).

Specific software for measurement of quality improvement is being developed. OMAX+ is a microcomputer based quality improvement measurement system. This system is an enhanced and modified version of the Objective Matrix Approach for productivity and quality measurement. It is specifically designed for companies that are implementing and using Total Quality Control (TQC) and provides a tool to quantify and track quality improvement. OMAX+ is described in Safford, Gobeli & Suen (1990).

The quality of the data is critical in any measurement system. Data quality, with e.g. dimensions such as timeliness, accuracy and completeness, is becoming an increasing important research area as the impact of unreliable data is realised. Fox, Levitin & Redman (1994) have laid a basis for the study of data quality. They discuss the four most important dimensions of data quality: accuracy, completeness, consistency and currentness as well as other related dimensions. They also discuss the five approaches to defining "data" and propose an approach within which data quality can be addressed.

Keith (1994: 31) conclusively remarks: *Meeting QIS objectives can result in a new synergy between customers and systems personnel. By working together toward common goals and taking advantage of the systems group's resources and services, people can achieve higher productivity levels and improved product and process quality. This, in turn, enhances competitive advantage and positively influences the bottom line.*

This summarises the ultimate impact of a good quality information system.

3.2.8 THE ROLE OF MEASUREMENT IN THE MALCOLM BALDRIGE NATIONAL QUALITY AWARD

The Malcolm Baldrige National Quality Award for businesses in the United States of America was institutionalised in 1987. The award serves the purpose of quality by giving awards to top quality companies and, in addition, the set of criteria used in evaluating the companies are also used internally by companies to do self-assessment.

The Baldrige criteria consist of a hierarchy of items: 7 categories, 32 examination items, and 99 areas to address (Brown 1991).

The seven categories are: leadership, information and analysis, strategic quality planning, human resource use, quality assurance of products and services, quality results and customer satisfaction (Brown 1991).

Two key factors that appear throughout the criteria (Brown 1991) are:

1) measurement
2) management by data rather than by experience or intuition.

The role of these two factors within six of the seven categories are summarised as follows:

In the category on **Information and Analysis:**

Items that are examined cover the following aspects:

1) The type of data that has to be collected to measure quality. The collection of the right data is important. Organisations sometimes measure indices that customers don't care about.
2) Benchmarking and competitive comparisons.
3) The use of the collected data. Is it really used by top management for decision-making?

In the **Strategic Quality Planning** category:

1) Quality and strategic goals need to be integrated into short and long-term business plans. Strategies for achieving the goals must be described.

In the **Human Resources** category:

1) Compensation and recognition programs that reward employees' quality improvement

efforts. Companies have difficulty in figuring out how to tie quality to reward systems. Most compensation plans do nothing to reward quality.

2) Evaluation of training effectiveness by testing.

In the **Quality Assurance of Products and Services** category:

1) Quality control in procurement, human resources, materials, marketing and sales, and other support departments.
2) Supplier quality - supplier training, certification and recognition programs.

In the **Quality Results** category:

1) The data for this category is data that are collected on products and services - e.g. "defects, rework, scrap, cycle time and delivery deadlines" (Brown 1991: 37).
2) Baldrige examiners look for positive data trends. Of importance is the slope as well as the degree to which results (in terms of quality) have been sustained.

In the **Customer Satisfaction** category (only external customer satisfaction is assessed):

1) A system for gathering customer-requirement data needs to be in place.
2) Different methods can be used for gathering data, e.g. interviews, telephone calls, surveys, etc. Finalist and award winners have measurable and specific standards relating to customer services.
3) A process for gathering customer complaints and resolving them in an efficient, timely manner.
4) The level of customer satisfaction of competitors is used for comparison of customer-satisfaction results.

In all the categories, measurement is an intrinsic part of the assessment for each category.

The winning companies of the Malcolm Baldrige National Quality Award in the United States of America (USA) in 1991 are mainly small, young, closely held companies. According to Davis

(1992: 39), *large publicly held companies with authoritarian cultures that must deal with continual carping of dissatisfied investors face different challenges* regarding quality management, namely:

1) a big cultural conversion and
2) they have to deal with a fixation on short-term financial results.

The three Malcolm Baldrige award winners of 1991 stress the following regarding measurement:

1) **Marlow Industries (Dallas, USA) - Small Business Winner**

Marlow was founded in 1973 with five people. Over the past two decades Marlow has averaged 15 percent annual growth and now employs 160 people (Davis 1992). It is the smallest business yet that won the Malcolm Baldrige Award.

Marlow Industries manufacture thermoelectric coolers - small solid state devices used for spot cooling in critical applications for telecommunications, aerospace and the military. Most of Marlow's products are custom-made, for customers who impose their own quality requirements on their suppliers. Marlow had to come up with a comprehensive quality system that would meet all of those requirements (Barrier 1992).

The process of deciding what to measure and how to measure it, remains a difficult task. Witzke, quoted in Barrier (1992), underlines the importance of a company needing to understand their processes and finding their key variables. Kendrick, Marlow's quality assurance manager, says: "With measurement of any kind, you need to look at what you want to do with the results" (Barrier 1992: 25). In Marlow, decisions on what should be measured were made on the basis of surveys of internal customers (Barrier 1992).

Davis (1992: 38) remarks: *Marlow asks customers to benchmark the requirements they would like them to meet. In turn, these requirements are passed on to suppliers so that they can help Marlow meet its customers' needs.*

"Supplier partnering" has been a critical area for improvement (Davis 1992). The company surveys purchasing, engineering and accounting performance of their suppliers. By means of informing their suppliers of a rating system and monitoring them on a regular basis, Marlow has improved supplier conformance and on-time delivery (Davis 1992).

Tailored "customer measures" to assess its own performance in each market segment are used by the Marlow company. *These measures are graphed, publicly displayed, and used to target further employee performance improvement* (Davis 1992: 38).

Marlow has also switched from product inspection to process control - i.e their quality people now look after the quality systems, set training standards and do audits (Barrier 1992).

From the above, it can be seen that the Marlow company has made extensive use of measurement. Furthermore, they integrated the information obtained from the suppliers, their own processes and their customers. This has proved to be of optimum benefit to their company.

2) The Solectron Corporation

Solectron Corporation is a contract electronic manufacturing company who supplies the computer peripherals and medical markets. Seventy percent of the company's business consists of printed circuit board assembly (Davis 1992).

They give their reason for success as the senior executive officer's leadership and commitment to "management by measurement".

Solectron has an extensive customer satisfaction measurement system (Davis 1992). Davis quoted Kennedy, VP for Quality Assurance and Technical Operations *We measure a lot of things every hour, every day, every week, every month* (Davis 1992: 38).

Solectron takes a sample of its customers weekly and mails the findings to its 70 best customers. Kennedy (Davis 1992: 38-39) says *Because customer feedback is often difficult to obtain we stimulate it by providing customers with this weekly feedback on what we see and ask them to* *provide us with feedback on what they see. This has proved valuable in adjusting our measures to our customer's measures and keeping on top of problems as they occur.*

Solectron also samples its customers for benchmarking data, i.e comparison data on satisfaction with competitive products and competitor field reliability. Comments and complaints are referred to the responsible managers and front line employees daily. Supplier performance is also monitored. The emphasis on quality and the focused measurement system have resulted in numerous improvements in many areas of company performance. At the same time, sales, profit, and earnings per share have also shown consistent improvement (Davis 1992).

Once again, a focused measurement system and the use of this information to improve business was stressed by the Solectron Company.

3) The Zytec Corporation

Zytec is the fifth largest United States manufacturer of power supplies for electrical equipment. It was founded in 1984 after a leveraged buy out from Control Data (Davis 1992).

Zytec has introduced quality justifications for capital equipment/investment instead of discounted cash flow and return on investment in financial management and performance reporting.

Zytec, Solectron and Marlow Industries have used both internally and externally focused measures to determine supplier quality, customer satisfaction, process control and improvement.

3.2.8.1 Critic against the Award

Critic on the program's focus is that its emphasis is almost exclusively on the internal quality process, to the exclusion of the systemic factors such as profitability and productivity. Impact on profitability and productivity has only lately been added and in a tangential way (Benson 1992). Harari (1993) calls the award counter productive because it reinforces the internal preoccupation. He notes that only 250 out of a possible 1000 points are allocated to the actual results of quality efforts. Schaffer and Thomson (1992) also criticise the award from a result point of view. They argue that companies are given high marks for outstanding quality processes without demanding that current products and services be equally outstanding.

However, critics admit that the problem of implementation are minor compared to what it has helped the United States of America do in terms of focusing management attention to the importance of quality as a strategic issue. In a survey among US businesses, Knotts, Parrish & Evans (1993) came to the same conclusion but add that the service and the industrial sectors have some differing views regarding the criteria.

The value of the database from the Internal Quality Study is that it put the Baldrige criteria into a performance context. It allows Baldridge executives to react in a dynamic way by integrating new concepts into the program based on the empirical evidence that certain management practices do indeed lead to measurable improvements for companies in certain performance positions (Benson 1992).

3.3 MEASURING QUALITY MANAGEMENT

3.3.1 INTRODUCTION

The measurement of quality management in organisations is considered to be an additional dimension of the measurement of quality at the strategic level and is therefore included in the dissertation.

The research by Saraph et al. (1989), Benson et al. (1991) and Flynn et al. (1994) is, to the author's knowledge, the only comprehensive studies on the subject of measuring quality management to date and have provided a foundation for research in this area.

Saraph et al. (1989) have developed an instrument for the measurement of the critical factors of quality management. It is an attempt to provide measures for organization-wide quality management. It can be used to obtain a profile of the quality practices within an organisation. It can also be used as an instrument for assessing the critical factors and identifying areas for improvement. Benson et al. (1991) have also used this instrument to test the effect of organisational context on quality management by means of an empirical study. Recently, Flynn et al. (1994) built on the work by Saraph et al. (1989), giving a framework for quality management research and providing an associated measurement instrument. These contributions will now be described and discussed.

3.3.2 THE EIGHT CRITICAL FACTORS OF QUALITY MANAGEMENT

Saraph et al. (1989) have identified eight critical factors from the literature. The factors, as well as an explanation, from Saraph et al. (1989: 818) are:

1. ***The role of management leadership and quality policy***

   Acceptance of quality responsibility by General Managers and department heads. Evaluation of top management on quality. Participation by top management in quality improvement efforts. Specificity of quality goals. Importance attached to quality in relation to cost and schedule. Comprehensive quality planning.

2. ***Role of the quality department***

   Visibility and autonomy of the quality department. The quality department's access to top management. Use of quality staff for consultation. Coordination between quality department and other departments. Effectiveness of the quality department.

3. ***Training***

   Provision of statistical training, trade training, and quality-related training for all employees.

4. ***Product/service design***

   Thorough scrub-down process. Involvement of all affected departments in design reviews. Emphasis on producibility. Clarity of specifications. Emphasis on quality, not roll-out schedule. Avoidance of frequent redesigns.

5. ***Supplier quality management***

Fewer dependable suppliers. Reliance on supplier process control. Strong interdependence of supplier and customer. Purchasing policy emphasizing quality rather than price. Supplier quality control. Supplier assistance in product development.

6. ***Process management***

Clarity of process ownership, boundaries, and steps. Less reliance on inspection. Use of statistical process control. Selective automation. Fool-proof process design. Preventative maintenance. Employee self-inspection. Automated testing.

7. ***Quality data and reporting***

Use of quality cost data. Feedback of quality data to employees and managers for problem solving. Timely quality measurement. Evaluation of managers and employees based on quality performance. Availability of quality data.

8. ***Employee relations***

Implementation of employee involvement and quality circles. Open employee participation in quality decisions. Responsibility of employees for quality. Employee recognition for superior quality performance. Effectiveness of supervision in handling quality issues. On-going quality awareness of all employees.

3.3.3 THE MEASUREMENT INSTRUMENT

The process that has been used to develop measures of the critical factors of quality management was based on generally accepted psychological principles of instrument design.

Operational measures for the critical factors of organisation-wide quality management have been developed by Saraph et al. (1989). A total of 78 items were chosen after initial selection and testing (see Appendix A). The items were included in a questionnaire. A five-point interval rating scale was used for each item, namely

Extent or Degree of Current Practise is

Very low	Low	Medium	High	Very high
1	2	3	4	5

For each critical factor, the actual level of practise can be represented by the average of the measurement item ratings for that factor. A vector of the averages for the eight factors can be used as a profile of the business unit's actual level of quality management.

3.3.4 ANALYSIS OF THE CRITICAL FACTOR MEASURES

3.3.4.1 Reliability

The reliability of the empirical measurements were assessed by the internal consistency method. The internal consistency of a set of measurement items refer to the degree to which a set of items are homogeneous. A reliability coefficient (Cronbach's alpha) was used to estimate internal consistency. Cronbach's alpha is *computed for a scale based on a given set of items.(... the scale score for all measures in this case is the mean of the item scores.) It can also be calculated for any subset of the items. It is therefore possible to identify the subset of items that has the highest reliability coefficient. The scale constructed from that subset is likely to be the best with regards to internal consistency* (Saraph et al. 1989: 820). A reliability coefficient of 0.7 or more are considered adequate.

Saraph et al. (1989) performed an internal consistency analysis using the Statistical Package for the Social Sciences (SPSS) reliability program. The scales (measures) that they developed were judged reliable.

3.3.4.2 Detailed item analysis

A method developed by Nunally [reported in Saraph et al. (1989)] was used to evaluate the assignment of items to scales. The method considers the correlation of each item with each scale. Specifically, the item-score to scale-score correlations are used to determine if an item belongs to the scale as assigned, belongs to some other scale, or if it should be eliminated. If an item does not correlate highly with any of the scales, it is eliminated (Saraph et al. 1989).

Saraph et al. (1989: 821) found that *all items have high correlation with the scales to which they were originally assigned to, relative to all other scales.* It was thus concluded that all items had been appropriately assigned to scales.

3.3.4.3 Validity

The validity of a measure *refers to the extent to which it measures what is intended to be measured* (Saraph et al. 1989: 823). Three different types of validity are considered:

1) content validity
2) criterion-related validity and
3) construct validity.

According to Saraph et al. (1989: 823), *A measure has content validity if there is general agreement among the subjects and researchers that the instrument has measurement items that cover all aspects of the variable being measured. Thus, content validity depends on how well the researchers created measurement items to cover the domain of the variable being measured.*

Content validity is subjectively judged by the researchers. Saraph et al. (1989: 23) argues that their measures have content validity as it was based on *an exhaustive review of the literature and detailed evaluations by academics and practising managers.* Their pretest subjects also agreed that the items represent the factors well.

Criterion-related validity is concerned *with the extent to which a measuring instrument is related to an independent measure of the relevant criterion. ... The eight measures of quality management in a business unit have criterion-related validity if these measures (collectively) are highly and positively correlated with quality performance in a business unit. In other words, these measures jointly should account for the performance of the business unit with respect to the quality of its products or services* (Saraph et al. 1989: 823).

The criterion-related validity was evaluated by studying the computed multiple correlation coefficient for the eight measures (collectively) and a measure of business unit quality performance. The measure used for quality performance is explained as follows in Saraph et al. 1989: 823): *Each manager was asked to rate (on a 5-point scale) the quality performance of their division for the past three years, as well as customer satisfaction with quality for the past three years. These two ratings were averaged to form a single measure of quality performance. This subjective measure was chosen over an objective measure because of the difficulty in identifying and obtaining an objective measure that would be appropriate for the different sizes and types of businesses in the sample.*

Saraph et al. (1989) found a multiple correlation coefficient of 0.8, which indicates that the eight measures (jointly considered) have a high degree of criterion-related validity.

A measure has construct validity if "it measures the theoretical construct or trait that it was designed to measure" (Saraph et al. 1989: 823). Factor analysis of the measurement items of each of the eight critical factors was used to evaluate the construct-validity of each critical-factor measure. The factor matrices showed that the items in seven of the eight measures formed a single factor. This can be used as tentative evidence of construct validity for these seven measures. Consideration should be given to split the process management items (where two factors were formed) into two separate constructs (Saraph et al. 1989).

3.3.5 INITIAL CONCLUSIONS

Saraph et al. (1989: 824) conclude: *The quality literature provides little guidance concerning how to measure any of the proposed critical factors of quality management. This paper successfully developed on this instrument that can be used to evaluate quality management in either the manufacturing or service organizations. The measures proposed were empirically based and shown to be reliable and valid ... Specification and measurement of the critical factors of quality management permit managers to obtain a better understanding of quality management practices ... Managers can use the instrument reported here to evaluate the perceptions of quality management in their organizations. These measurements can help decision makers identify those areas of quality management where improvements should be made. Also, comparisons of* *different organizations or divisions can be made to help prioritize quality management efforts.*

The instrument can thus be regarded as an external evaluation instrument for quality management.

3.3.6 THE EFFECT OF ORGANISATIONAL CONTEXT ON QUALITY MANAGEMENT

Benson et al. (1991: 1108) discuss a system-structural view of quality management. They remark: *The system-structural view explicitly considers the organization's external context and its impact on the organization. With quality problems being driven by external factors such as customer demands, competitive pressures, and government regulations, the system-structural view is particularly helpful in explicating a theory of quality management.*

A System-Structural View of Quality Management as well as a System-Structural View of Quality Management modified to reflect aspects of the managerial problem-solving process are given in figures 3.2 and 3.3 (Benson et al. 1991: 1109).

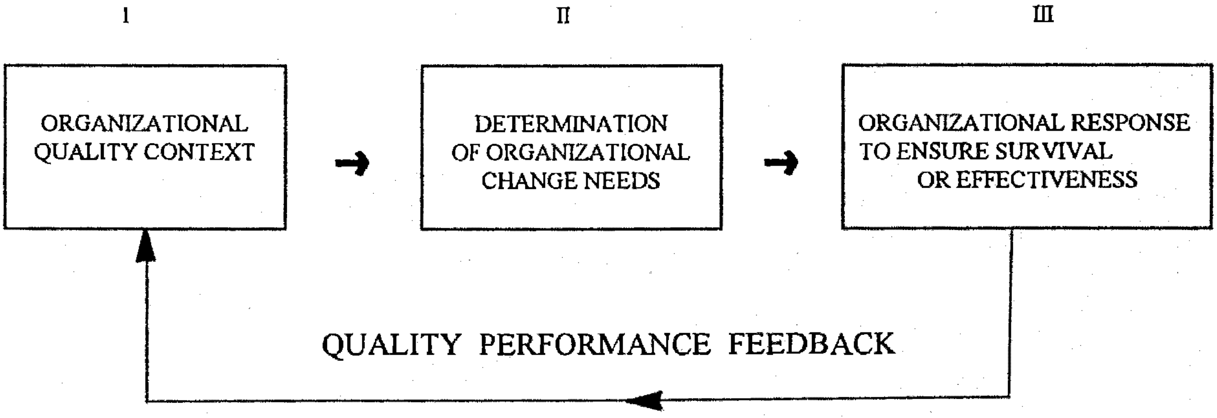

I
II
III
ORGANIZATIONAL QUALITY CONTEXT
DETERMINATION OF ORGANIZATIONAL CHANGE NEEDS
ORGANIZATIONAL RESPONSE TO ENSURE SURVIVAL OR EFFECTIVENESS
QUALITY PERFORMANCE FEEDBACK

Figure 3.2 A System-Structural View of Quality Management

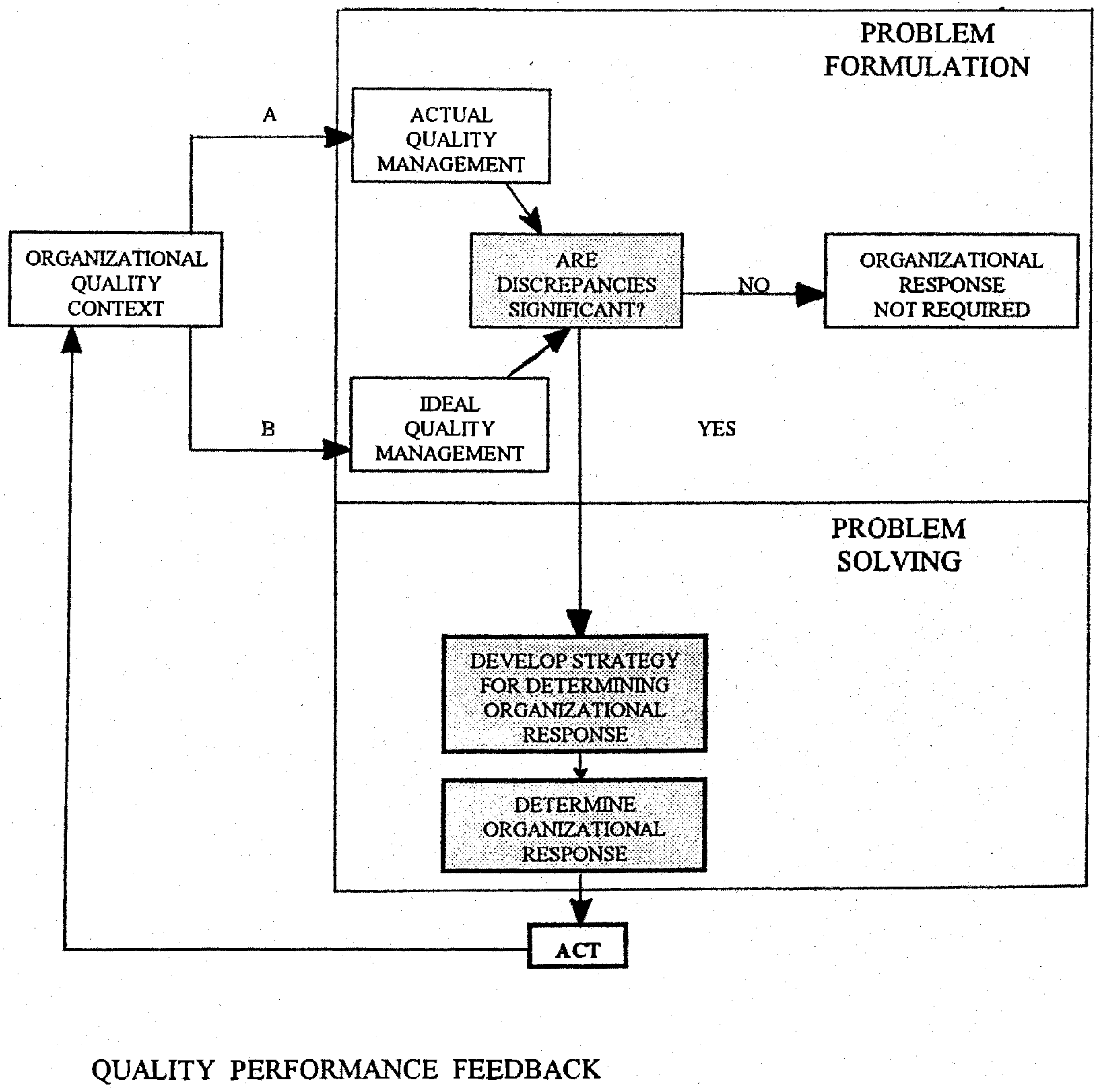

PROBLEM FORMULATION
A
ACTUAL QUALITY MANAGEMENT
ORGANIZATIONAL QUALITY CONTEXT
ARE DISCREPANCIES SIGNIFICANT?
NO
ORGANIZATIONAL RESPONSE NOT REQUIRED
IDEAL QUALITY MANAGEMENT
B
YES
PROBLEM SOLVING
DEVELOP STRATEGY FOR DETERMINING ORGANIZATIONAL RESPONSE
DETERMINE ORGANIZATIONAL RESPONSE
ACT
QUALITY PERFORMANCE FEEDBACK

Figure 3.3 The System-Structural View of Quality Management Modified to Reflect Aspects of the Managerial Problem-Solving Process

The hypotheses tested are:

1) *managers' perceptions of actual quality management are influenced by organizational contextual variables*

2) *managers' perceptions of ideal quality management are not affected by organizational contextual variables* (Benson et al. 1991: 1110).

Ideal quality management is *a business unit manager's beliefs concerning what quality management should be in the business unit* (Benson et al. 1991: 1110).

Actual quality management is *the manager's perception of the current practise of quality management in the unit (business)* (Benson et al. 1991: 1110).

Both ideal and actual quality management were measured in terms of the eight comprehensive, critical factors described in Saraph et al. (1989).

Organisational quality context is *the business unit manager's quality environment* (Benson et al. 1991: 1110).

A summary of the organisational quality context variables that were considered in Benson et al. (1991: 1113) are given:

1) *Managerial Knowledge*
2) *Corporate Support for Quality*
3) *Product/Process Contextual Variables*
    - *Rate of Product/Process Change*
    - *Proportion of Products/Services Purchased Outside*
    - *Degree of Manufacturing Content*
    - *Extent of Batch vs. Continuous process*
    - *Product Complexity*

4) *Past Quality Performance*

- *Past 3 Years' Quality Performance*
- *Degree of Customer Satisfaction for Past 3 Years*

5) *Marketplace Contextual Variables*

- *Degree of Competition*
- *Extent of Entry Barriers*
- *Extent of Customer Quality Demands*
- *Extent of Regulatory Quality Demands*

6) *Company Size (Large, Medium, Small)*

7) *Company Type (Manufacturing or Service)*

8) *Manager Type (General Manager or Quality Manager)*

A factor analysis, using the SPSS package, was used to reduce the 26 organisational quality context measurement items (see Appendix B) to a manageable and meaningful set of variables. Four factors were identified that accounted for 78% of the total variance of the original 26 items. The four factors are: Corporate Support for Quality, Managerial Knowledge, Past Quality Performance and the last factor comprises two of the four items in the Marketplace Environment Section, namely "quality demands of customers" and "regulatory and legal requirements on quality". Thus, 19 of the original 26 measurement items were reduced to four factors. The seven items that did not load on any single factor were treated as separate variables.

The organisational quality context variables selected for subsequent analysis are given in table 3.2.

VARIABLE	DESCRIPTION
X1	Managerial Knowledge
X2	Corporate Management Support for Quality
X3	Degree of Competition
X4	Barriers to entry in the industry
X5	External Quality Requirements
X6	Rate of Product/Process Change
X7	Proportion of Products/Services Purchased Outside.
X8	Degree of Manufacturing Content
X9	Extent of Batch vs. Continuous Process
X10	Product Complexity
X11	Past Quality Performance
MANAGER TYPE	Two levels: General Manager and Quality Manager
COMPANY SIZE	Three levels: Large, medium and small
COMPANY TYPE	Two levels: Manufacturing and service

Table 3.2 Organisational quality context variables

Multivariate analysis of covariance (MANCOVA) was used to examine the effects of company size, company type and manager type on ideal quality management. It was found that none of the three factors were significant. None of the factors were thus useful for explaining variation. Consequently, they were not included in the canonical correlation analysis of the relationship between the ideal quality management variables and the organisational quality context variables.

The same analysis was performed using the actual quality management variables as the dependent variables. Neither company size nor manager type were significant, but company type (Manufacturing or Service) was. The canonical correlation analysis of the relationship between the actual quality management variables and the organizational quality context variables was then applied separately for the subsample of managers from service firms and the subsample from manufacturing firms.

Canonical correlation analysis was used to investigate the hypotheses stated. The particular relationships analysed were:

1) *between the set of seven variables that describe actual quality management and the set of quality context variables, separately for the manufacturing and service subgroups and*
2) *between the seven variables that describe ideal quality management and the quality context variables for the full sample of managers* (Benson et al. 1991: 1118).

The significance of the canonical correlation coefficients was tested using Bartlett's chi square test. The results support the hypothesis that managers' perceptions of actual quality management are influenced by organisational contextual variables. The most influential contextual variables are corporate support, past quality performance, and, based on the MANCOVA results, type of firm (manufacturing or service).

Concerning the second hypothesis that managers' perceptions of ideal quality management are not influenced by their organisational context, the results indicate the alternative. It seems that managers' beliefs concerning ideal quality management are apparently not context-free. The data indicates that *all seven aspects of ideal quality management are influenced by organizational context and the important contextual variables are manager's knowledge, corporate support for quality, external quality requirements and product complexity* (Benson et al. 1991: 1120).

Past quality performance was identified as an influential contextual variable in both the service and manufacturing sector regarding actual quality management. This result according to Benson et al. (1991: 1120): *confirms the need for the model's quality performance feedback loop. The loop indicates that current managerial actions affect the organisation's quality context and, thus, management's future perceptions of actual and ideal quality management.*

An interesting result is the fact that actual quality management in manufacturing organisations was affected by both internal contextual factors (corporate support for quality, past quality performance, and management knowledge) and external contextual factors (extent of entry barriers and external quality demands). In the service subsample, however, only internal factors (corporate support for quality, past quality performance, and product complexity) were correlated with actual quality management. More research on this difference between manufacturing and service is required.

The importance of top management leadership and support for the successful implementation of quality management has been confirmed by this study.

Product complexity affects service firms. This is probably because service industries are more diverse and apply different quality management practices in different types of product environments (e.g. airlines, insurance, utilities, etc.)

Furthermore, although not context-free, the study does support the idea that beliefs concerning ideal management do not systematically differ over a wide range of contextual variables. Benson et al. (1991: 1122) also found that *perceptions of ideal quality management are more influenced by 'thought leaders' at corporate levels, or by external requirements, than by size of company, industry, type of manager, or product characteristics. There appears to be a strong impact of knowledge and leadership on the perceptions of ideal quality management.*

The findings suggest that knowledge of organisational quality context is useful for explaining and predicting quality management practise. Benson et al. (1991) recommend that future work should focus on explaining the processes that managers use to formulate and solve quality management problems.

3.3.7 A FRAMEWORK FOR QUALITY MANAGEMENT RESEARCH AND AN ASSOCIATED MEASUREMENT INSTRUMENT

Research regarding quality management should firstly specify the important dimensions of quality management (Flynn et al. 1994). These dimensions must be measured, and it must be determined that the measures are reliable and valid. Only then can the effect of quality management on performance be determined. Recent literature emphasized the measurement of quality performance (output of process) and not that of quality management (input of process). Very little empirical research has been focused on quality management practises (Flynn et al. 1994).

The paper of Flynn et al. (1994) builds on the work of Saraph et al. (1989) but differs with respect to the following:

1) Saraph et al. (1989) measure managers' perceptions of the eight critical factors at the **business unit level.** Flynn et al. (1994) design their instrument to measure at the **plant (manufacturing environment) level.**
2) Saraph et al's (1989) instrument is designed for use by the **quality** and **general managers,** measuring their **perception** of the degree of quality practises. Flynn et al. (1994) have different instruments for different groups, e.g. there exist separate instruments for direct laborers, supervisors, production and inventory managers, the process engineer and human resources manager.
3) The study of Flynn et al. (1994) is more manufacturing-specific.
4) The literature bases for the two studies differ. Saraph et al. (1989) use the theoretical work of acknowledged quality experts (Deming, Juran, Crosby, etc) while Flynn et al. (1994) concentrate on practitioner and empirical literature coming from actual quality management practises in Japan and the USA. Interestingly enough, Flynn et al. (1994) note that both studies led to similar dimensions.

Flynn et al. (1994) advise that both of the instruments proposed could be useful when studying the impact of quality management practises on performance.

Flynn et al. (1994: 342) define **quality management** as follows: ***An integrated approach to achieving and sustaining high quality output, focusing on the maintenance and continuous improvement of processes and defect prevention at all levels and at all functions of the organization, in order to meet or exceed customer expectations.***

Flynn et al. (1994) further discuss the role of quality management as a key element within the World Class Manufacturing approach. They identify the following seven dimensions to be the core dimensions of quality management:

1) top management support
2) quality information

3) process management
4) product design
5) workforce management
6) supplier involvement
7) customer involvement.

They note the relationship between their dimensions and the categories of the Baldrige Award (described in 3.2.8), stressing that each of their dimensions can be directly linked to the categories.

The conceptual foundation for their proposed measurement instrument was based on a literature review and a series of plant visits.

Flynn et al. (1994) have developed the measurement instrument for quality management practises, concentrating on the aspects of reliability and validity of the instrument. They did a measurement analysis by firstly assessing the instrument's reliability (the ability of the scales of the instrument to consistently yield the same response) and then assessing validity (the scale's ability to measure what it is set to measure). They assessed three dimensions of validity: content validity, construct validity and criterion-related validity.

Detailed information regarding the instrument's items and development, the sample selection and the reliability and validity analysis can be found in Flynn et al. (1994). The iterative process used to determine the final version of the scales are also included in their article.

Flynn et al. (1994) conclude that their results provide tentative evidence that the instrument they presented is reliable and valid. They advise that further work is needed to:

1) refine the instrument and increase its alpha values (reliability)
2) focus on the relationship between quality management practises and quality performance and overall plant performance through analytical work with the instrument
3) generalize the results to industries beyond those tested
4) determine the appropriateness of the instrument for other countries and cultures
5) improve the testing of criterion-related validity by the collection of additional objective

measures of the criteria

6) develop a more comprehensive instrument that would permit plant level as well as divisional and corporate level use to assess quality management practises (i.e. examination of the effectiveness of top-down versus bottom-up quality management strategies) and

7) include customer perceptions of quality performance.

Their final concluding remark (Flynn et al. 1994: 362) echoes once again the importance of good measurement practises in an organisation: *Reliable and valid scales are an important means of self-assessment for an organisation, and should provide a key input into planning efforts, providing a factual basis for making decisions in areas which are often difficult to quantify. The use of reliable and valid measurement scales may be a vital part of benchmarking an organisation's performance against referent organisations.*

3.3.8 CONCLUSION

The critical factors (dimensions) that are similar in both studies (Flynn et al. (1994); Saraph et al. (1989)) are:

1) the role of management leadership and quality policy (top management support)
2) product/service design (product design)
3) supplier quality management (supplier involvement)
4) process management (process management)
5) quality data and reporting (quality information).

Saraph et al. (1989) separate training as a critical factor from employee relations. In Flynn et al. (1994) training and employee relations is part of the workforce management factor.

It is interesting to note that the dimension which is included in Saraph et al. (1989) but not in Flynn et al. (1994) is the role of the quality department. It is probably due to the fact that the literature surveyed for Saraph's study concentrate on quality practises from within the company. In Flynn's study, concentrating on practical and empirical work, this has not surfaced as a critical factor. Instead, the role of customer involvement has surfaced. This aspect is very important and has been discussed in 3.2.4.

Saraph et al. (1989), Benson et al. (1991) and Flynn et al. (1994) have thus provided a basis for research into this aspect which should be further pursued.

3.4 QUANTITATIVE STRUCTURES FOR PROCESS IMPROVEMENT

> *A young man carrying a violin case stopped a cab driver in New York City and asked him, "How do I get to Carnegie Hall?" The cab driver answered, "Kid, practise, practise, practise!"*
>
> Bossert (1991: 51)

Continuous quality improvement (CQI) needs to be measured in order to determine whether any improvement did take place as a result of certain improvement actions taken by the organisation. The author will give a brief overview of innovative structures and techniques currently used in industry as a vehicle to support, control and measure improvement.

The quality management maturity grid, cleanroom software engineering, software factories, quality function deployment, the seven planning tools, benchmarking and the ISO 9000 series of standards will be briefly described as to what each constitute. Each of these is a comprehensive subject and as such, no attempt will be made to discuss detail. It is written to create an awareness of the vast number of quantitative structures and techniques that have been established over the past few years and that can be applied successfully in business. Statistical Process Control (SPC), an intrinsic part of the process of measuring quality and a very important set of techniques, is acknowledged but will not be discussed in this dissertation.

The establishment of an infrastructure to accommodate the necessary data collection, analysis and feedback is a critical element in the success of the use of any technique or structure. These elements will be described in the contents of software metric programs and software cost estimation.

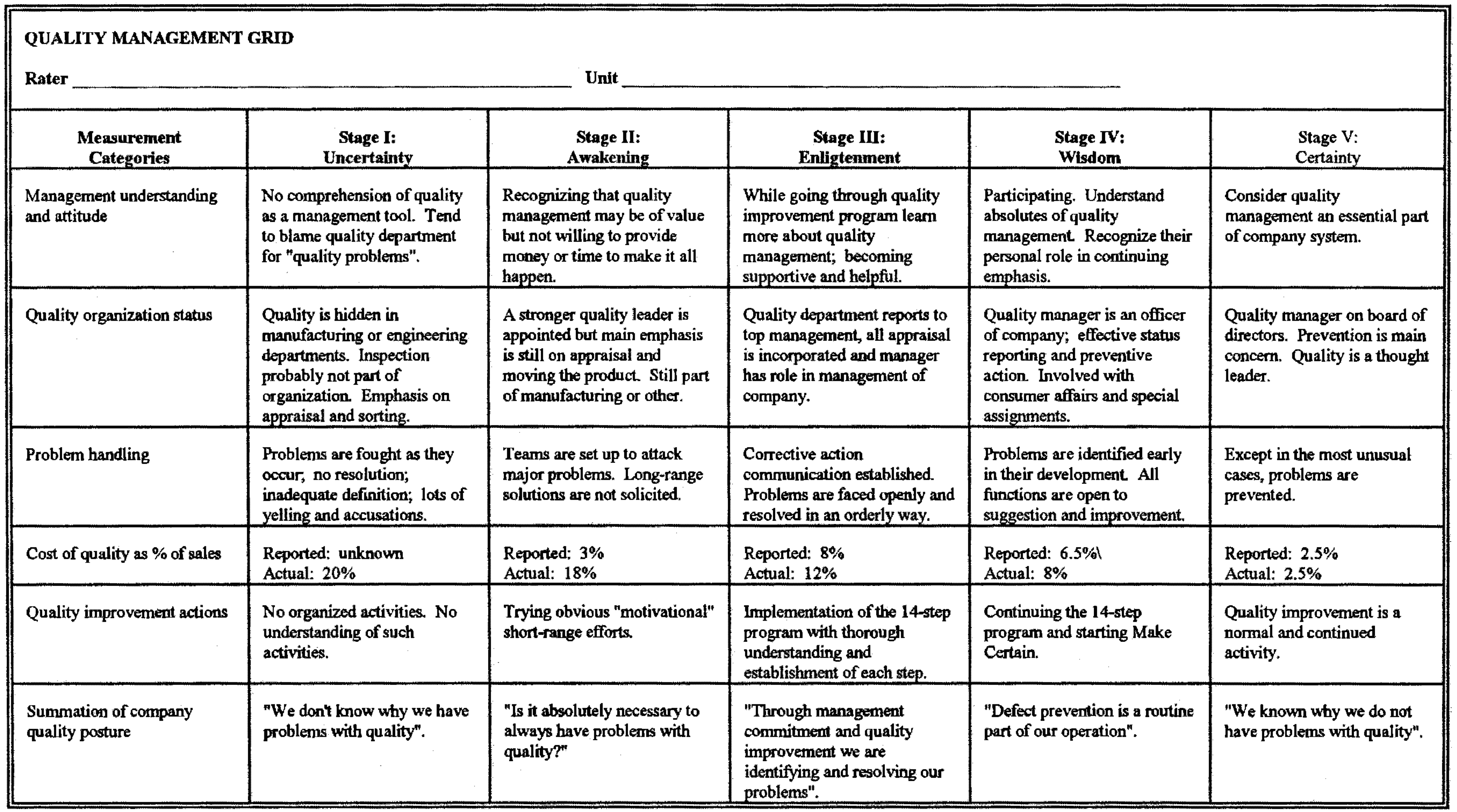
3.4.1 THE QUALITY MANAGEMENT MATURITY GRID

Crosby (1979) developed the quality management maturity grid for organisations. He recognises the fact that quality management has been seen as subjective, and therefore difficult to define and measure. He blames this on the fact that people see it as a **result-oriented task** rather than a **planning operation.** Using his quality management maturity grid, a manager should be able to classify his/her operation's quality state. As Crosby (1979: 27) remarks: *All that is required is knowing what is going on.*

The grid is divided into five stages of maturity:

1) uncertainty
2) awakening
3) enlightenment
4) wisdom
5) certainty

The following figure comes from Crosby (1979: 38-39) and explains the stages according to six measurement categories:



QUALITY MANAGEMENT GRID

Rater ______________________ Unit ______________________

Measurement Categories	Stage I: Uncertainty	Stage II: Awakening	Stage III: Enligtenment	Stage IV: Wisdom	Stage V: Certainty
Management understanding and attitude	No comprehension of quality as a management tool. Tend to blame quality department for "quality problems".	Recognizing that quality management may be of value but not willing to provide money or time to make it all happen.	While going through quality improvement program learn more about quality management; becoming supportive and helpful.	Participating. Understand absolutes of quality management. Recognize their personal role in continuing emphasis.	Consider quality management an essential part of company system.
Quality organization status	Quality is hidden in manufacturing or engineering departments. Inspection probably not part of organization. Emphasis on appraisal and sorting.	A stronger quality leader is appointed but main emphasis is still on appraisal and moving the product. Still part of manufacturing or other.	Quality department reports to top management, all appraisal is incorporated and manager has role in management of company.	Quality manager is an officer of company; effective status reporting and preventive action. Involved with consumer affairs and special assignments.	Quality manager on board of directors. Prevention is main concern. Quality is a thought leader.
Problem handling	Problems are fought as they occur; no resolution; inadequate definition; lots of yelling and accusations.	Teams are set up to attack major problems. Long-range solutions are not solicited.	Corrective action communication established. Problems are faced openly and resolved in an orderly way.	Problems are identified early in their development. All functions are open to suggestion and improvement.	Except in the most unusual cases, problems are prevented.
Cost of quality as % of sales	Reported: unknown Actual: 20%	Reported: 3% Actual: 18%	Reported: 8% Actual: 12%	Reported: 6.5%\ Actual: 8%	Reported: 2.5% Actual: 2.5%
Quality improvement actions	No organized activities. No understanding of such activities.	Trying obvious "motivational" short-range efforts.	Implementation of the 14-step program with thorough understanding and establishment of each step.	Continuing the 14-step program and starting Make Certain.	Quality improvement is a normal and continued activity.
Summation of company quality posture	"We don't know why we have problems with quality".	"Is it absolutely necessary to always have problems with quality?"	"Through management commitment and quality improvement we are identifying and resolving our problems".	"Defect prevention is a routine part of our operation".	"We known why we do not have problems with quality".

Figure 3.5 Crosby's Grid

Crosby (1979) states that, by reading the information in each block, one is able to identify one's own situation. The following remark puts the use of the grid in perspective: *The grid is at its best when used to project a view of the company that all involved can accept ... It also provides a continual source of direction concerning what needs to be done next* (Hughes 1985: 18).

Apart from the software industry the quality management maturity grid has been minimally used. The only reports found on using the grid in other environments were in Hughes (1985); Lee and Willis (1988) and Sweet (1983).

Hughes (1985) applies the grid to safety management, calling it the safety management maturity grid. He stresses the importance of having a **quantitative** yardstick. He describes the grid as applied to safety management and reduces the measurement categories from six to five to cater for the safety and health situation in an organisation.

Lee and Willis (1988) describe the use of the quality management maturity grid to determine the level of quality/productivity that each business unit has achieved and to check overall progress after the first phase of the quality improvement program. The business units are units in the Manufacturers and Traders Trust Company, a commercial bank situated in Buffalo, New York, USA.

Sweet (1983) describes a purchasing management maturity grid, developed at the Harris Company (USA), based on Crosby's grid. It is used to enable purchasing management to determine whether their departments have reached their full potential regarding efficiency, professionalism, and status.

The quality management maturity grid thus provides a way of continually measuring the quality management process. It differs from the method suggested by Saraph et al. (1989), discussed in 3.3.2, in the sense that it takes on a continuing process perspective and is not intended as a strict measurement instrument to measure quality management at one point in time.

Humphrey (1988) adapted Crosby's grid when defining his Software Process Maturity Framework as an instrument to characterize the capabilities of software development organisations. Humphrey (1988: 74) uses a process-orientation by describing the entire software development task as a process that can be “controlled, measured and improved”. He defines a software process as *that set of actions required to efficiently transform a user's need into an effective software solution* (Humphrey 1989: x).

The original assessment approach was developed to assist the USAF (United States Air Force)/ DoD (Department of Defence) software contractor evaluation methods. It was developed at the Software Engineering Institute (SEI) of the Carnegie-Mellon University, USA. A software process assessment method and a software capability evaluation method as well as a maturity questionnaire was used to determine maturity. It's name was later changed to *Capability Maturity Model for Software* (Paulk et al. 1993). This model, according to Paulk et al. (1993: 18), *presents sets of recommended practices in a number of key process areas that have been shown to enhance software-development and maintenance capability.*

The five maturity levels (Humphrey 1988) are:

1) initial
2) repeatable
3) defined
4) managed
5) optimizing

A summary of the five levels (Humphrey 1991) are shown in table 3.3.

LEVEL	CHARACTERISTIC	KEY CHALLENGE
1 INITIAL	(Ad hoc/ chaotic)	Project management Project planning Configuration management Software quality assurance
2 REPEATABLE	(Intuitive) Process dependent on individuals	Training Technical practises reviews, testing Process focus standards; process groups
3 DEFINED	(Qualitative) Process defined and institutionalised	Process measurement Process analysis Quantitative quality plans
4 MANAGED	(Quantitative) Measured Process	Changing technology Problem analysis Problem prevention
5 OPTIMIZING	Improvement feedback into process	Still human intensive process. Maintain organization at optimizing level.

Table 3.3 SEI Software Process Maturity Model

A comprehensive description of each maturity level can be found in Humphrey (1988) and Humphrey (1989). The framework thus helps organisations to assess themselves and identify the areas that need priority for improvement. The basic objective is to establish a controlled and measured process as a foundation for continuous improvement.

The SEI developed several aids to help in assessments, such as: *SEI-assisted assessments, assessment tutorials, self-assessments, SEI-licensed vendor assessments and capability evaluations* (Humphrey 1991: 263).

In the software industry, the Software Process Maturity Framework has been given a lot of attention and subsequently assessment of organisations is done world-wide (Humphrey 1991). It is also linked to the selection of metrics in a process maturity-based metrics approach (Pfleeger & McGowan 1990). They suggest the implementation of metrics to correspond with the maturity level of the organisation, i.e. the metrics are implemented step by step. In the **Initial** stage metrics need to be selected that can serve as a baseline for comparisons. The next stage, **Repeatable,** needs metrics focused on project management. In Stage 3, **Defined,** the metrics must measure the product during development. The **Managed** stage requires metrics that "capture characteristics of the development process itself to allow control of the process itself' (Pfleeger & McGowan 1990: 225). In the final stage, the metrics are process metrics with feedback loops to enable utilisation of metrics for changing the process. Pfleeger (1991b) also describes the use of process maturity as guidelines in the selection of CASE (Computer-Aided Software Engineering) tools. Rugg (1993) describes the use of the Capability Maturity Model to select a software contractor and stresses the usefulness of the evaluations for the organisation that are evaluated. The Capability Maturity Model (version 1.1) is discussed in Paulk et al. (1993). Figure 3.5 depicts the CMM model (Fenton & Whitty 1995: 4).

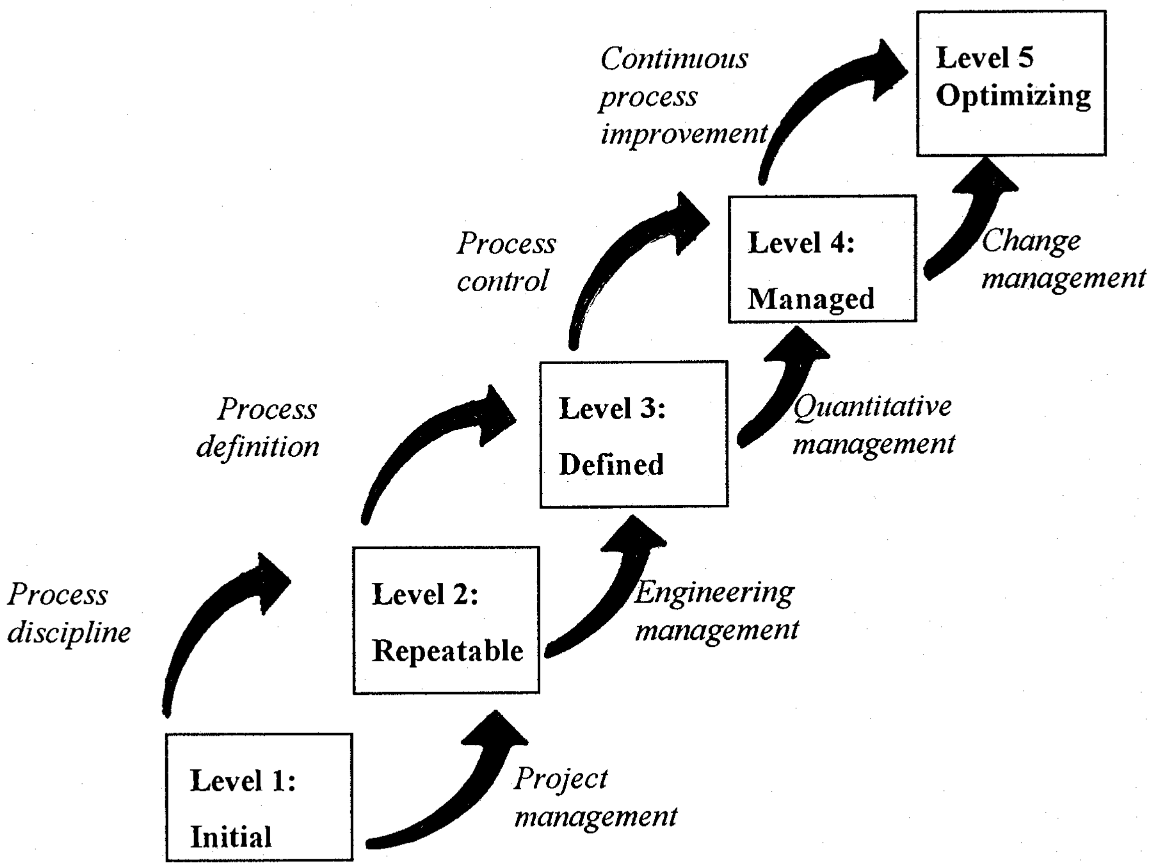


Figure 3.5 Capability Maturity Model (CMM)

Humphrey (1991) states that in a sample of ± 200 United States software development companies, it was found that over 80% were at the lowest level (initial) of maturity and most of the remaining companies were at level two (repeated).

In their article: "A critical look at software capability evaluations", Bollinger and McGowan (1991) comment on some serious flaws in the current grading system. They argue that the model fails to take a top-down perspective on how processes should be designed and optimized and it also fails to recognise the effects of different types of risks on the software process. The use of the evaluation (a single 85-question yes/no test) to accredit organisations responsible for developing all the software for the Defence Force of the United States of America, are questioned.

The Bootstrap approach, an alternative assessment approach, uses the maturity model as basis and will be subsequently described.

3.4.1.1 The Bootstrap approach

According to Koch (1993), there are two distinct groups of empirical software engineering research scientists: The first group, *relies on a tradition (originating from the age of elucidation of being able to measure software engineering in an absolutistic and rationalistic way) and is basically associated with the SEI.* The second group *does not believe in ordinal scales and is more interested in improving the software processes by self-referential improvement exercises* (Koch 1993: 391).

Underlying to the BOOTSTRAP approach is the Kaizen notion explained by Huda and Preston (1992: 10) as *KAIZEN is more akin to a philosophy and defies rigid definition; rather it is an amalgamation of interrelated principles which singly are inconsequential but combined become a powerful method of initiating improvement. Kaizen is a holistic approach to problem solving and its difference lies in being people-centred rather than system-centred. It recognizes the overriding importance of the human element and gives a new perspective to problem solving by way of minimizing conflict and of eliminating blame, so that people work together instead of individually towards goals.*

The ESPRIT[1] project, BOOTSTRAP, developed an assessment method which used the maturity model as the basis. They extended the original SEI questionnaire. The method also includes questions based on ISO 9000 quality standards and the European Space Agency's process model (Bootstrap: Europe's Assessment Method 1993). Their methodology thus describes the assessment process, determines the maturity level of an organisation, identifies the strengths and weakness (capability) and offers action plans for improvement.

BOOTSTRAP also differs from the SEI maturity model with respect to self-assessment of organisations. Bootstrap does not support self-assessment.

Two questionnaires (one for the whole software producing unit (SPU) and one for projects) are used to gather data. The questions are divided into three groups relating to

1) the organisation,
2) the methodology and engineering know-how and
3) technology transfer.

A five point scale is used (absent, weak, fair, extensive and non-applicable). The SEI maturity model only have yes/no categories for all their questions.

The BOOTSTRAP methodology also have five maturity levels but includes, in addition, quartiles within these levels. The maturity level is determined through an algorithm which allocate a certain maturity level *if the scores of answers from the questions within that level fits inside defined value limits* (Bootstrap: Europe's Assessment Method 1993: 94).

Twenty-one capability factors (a set of criteria which make up a SPU's or project's capability) are considered. A set of questions is devoted to each factor in the questionnaire. The maturity level is obtained by applying the "BOOTSTRAP" algorithm to the set of questions. It is important to note that this has nothing to do with the statistical bootstrap methodology, it only refers to the algorithm developed by the team members of the BOOTSTRAP project.

[1] European Strategic Programme of Research and Development in Information Technology

A detailed discussion about the BOOTSTRAP project's approach can be found in Koch (1993).

Several important quantitative issues within the 'BOOTSTRAP' approach, are still regarded as research issues by Koch (1993). They are:

1) the verification of the mountain climbing algorithm and the compatibility between the BOOTSTRAP calculation method and the SEI's method
2) the principles of statistics applied to the questionnaire have to be verified
3) the data collected have to be analysed for additional empirical evaluations
4) the correlation between a particular maturity level, the product quality achieved at that level
5) the productivity in relation to the quality achieved. This issue is also mentioned by (Bootstrap: Europe's Assessment Method 1993).

3.4.2 CLEANROOM SOFTWARE ENGINEERING

The cleanroom engineering approach involves the engineering of software under statistical quality control (Mills, Dyer & Linger 1987). The approach requires the specification of the functional behaviour of the software as well as its statistical usage. The first priority of this approach is the prevention of defects rather than the removal of defects. The second priority is to *provide valid statistical certification of the software's quality through representative-user testing at the system level* (Mills et al. 1987: 19). The aim is to deliver software with a known and certified mean time to failure (MTTF) (Dyer 1992).

The cleanroom approach requires stable specifications for the software as its basis. It also requires the *development of software in increments that permit realistic measurements of statistical quality during development, with provision for improving the measured quality by additional testing, by process changes (such as increased inspections and configuration control), or by both methods* (Mills et al. 1987: 21).

Mills et al. (1987) stress that no "best statistical measure" for software quality exist and that the measure chosen to represent software quality remains a judgement of business and management.

A basis for the certification of software quality was developed by Currit, Dyer and Mills (1986). The certification is given in terms of the *measured reliability over a probability distribution of usage scenarios in statistical testing* (Mills et al. 1987: 21).

Cleanroom engineering uses mathematical verification, done by people, to replace the interactive debugging of programs before release to statistical testing. Mathematical verification requires, according to Mills et al. (1987: 21) *precise specifications and formal arguments about the correctness with respect to those specifications.* Mills et al. (1987) argue, on the basis of experience with three projects that utilise the Cleanroom approach, that it produces software sufficiently robust to go to system testing without debugging.

A detail description of the cleanroom engineering approach can be found in the book: ***The cleanroom approach to quality software development*** by Michael Dyer (1992).

3.4.3 SOFTWARE FACTORIES

The term **software factories** refers to an approach of applying factory concepts to the management of software development. In his book, **Japan's Software Factories**, Cusumano (1991), describes the application of this approach by the Hitachi, Toshiba, NEC and Fujitsu companies. These companies have all *attempted the strategic management and integration of activities required in software production, as well as the achievement of planned economies of scope-cost reductions or productivity gains that come from developing a series of products within one firm (or facility) more efficiently than building each product from scratch in a separate project* (Cusumano 1991: 8).

Certain common elements in the factory approach regarding implementation across a series of similar projects (Cusumano 1991: 9) are:

1) *commitment to process improvement*
2) *product-process focus and segmentation*
3) *process quality analysis and control*
4) *tailored and centralized process R & D*

5) *skills standardization and leverage*

6) *dynamic standardization*

7) *system reusability*

8) *computer-aided tools and integration*

9) *incremental productivity/variety improvement.*

According to Cusumano (1991), the key lesson to be learned is the achievement of an effective balance of process efficiency and flexibility in the production of unique and customized software through the application of the principles of a software factory. He stresses the words of Yukio Mizuno (of NEC) that the software factory is essentially *a concept and not a thing; a philosophy that at least some software could be produced in a manner more akin to engineering and manufacturing than craft or cottage-industry practices* (Cusumano 1991: 443).

In Europe, the Eureka Software Factory (ESF) project was established as part of the Eureka programme. The prime goal is the establishment of software factories in practice in industry. An overview of the Eureka Software Factory is given in Thomas, Fernstroem and Hesse (1991).

3.4.4 QUALITY FUNCTION DEPLOYMENT

Quality function deployment (QFD) is described by Bossert (1991: 1) as follows: *Quality function deployment is a process that provides structure to the development cycle. This structure can be likened to the framework of a house. The foundation is customer requirements. The frame consists of the planning matrix, which includes items such as the importance rating, customer-perceived benchmarking, sales point, and scale-up factors. The second floor of the house includes the technical features. The roof is the trade-off of technical features. The walls are the interrelationship matrix between the customer requirements and the technical characteristic. Other parts can be build using things such as new technologies, functions, technical characteristics, processing steps, importance ratings, competitive analysis, and sales points. The components utilized are dependent on the scope of the project.*

The technique is also referred to as the **House of Quality** (Hauser & Clausing 1988). The foundation of these techniques is the belief that products should be designed to reflect the needs of the customer.

A basic matrix showing the various components is depicted in figure 3.6 (Bossert 1991: 7).

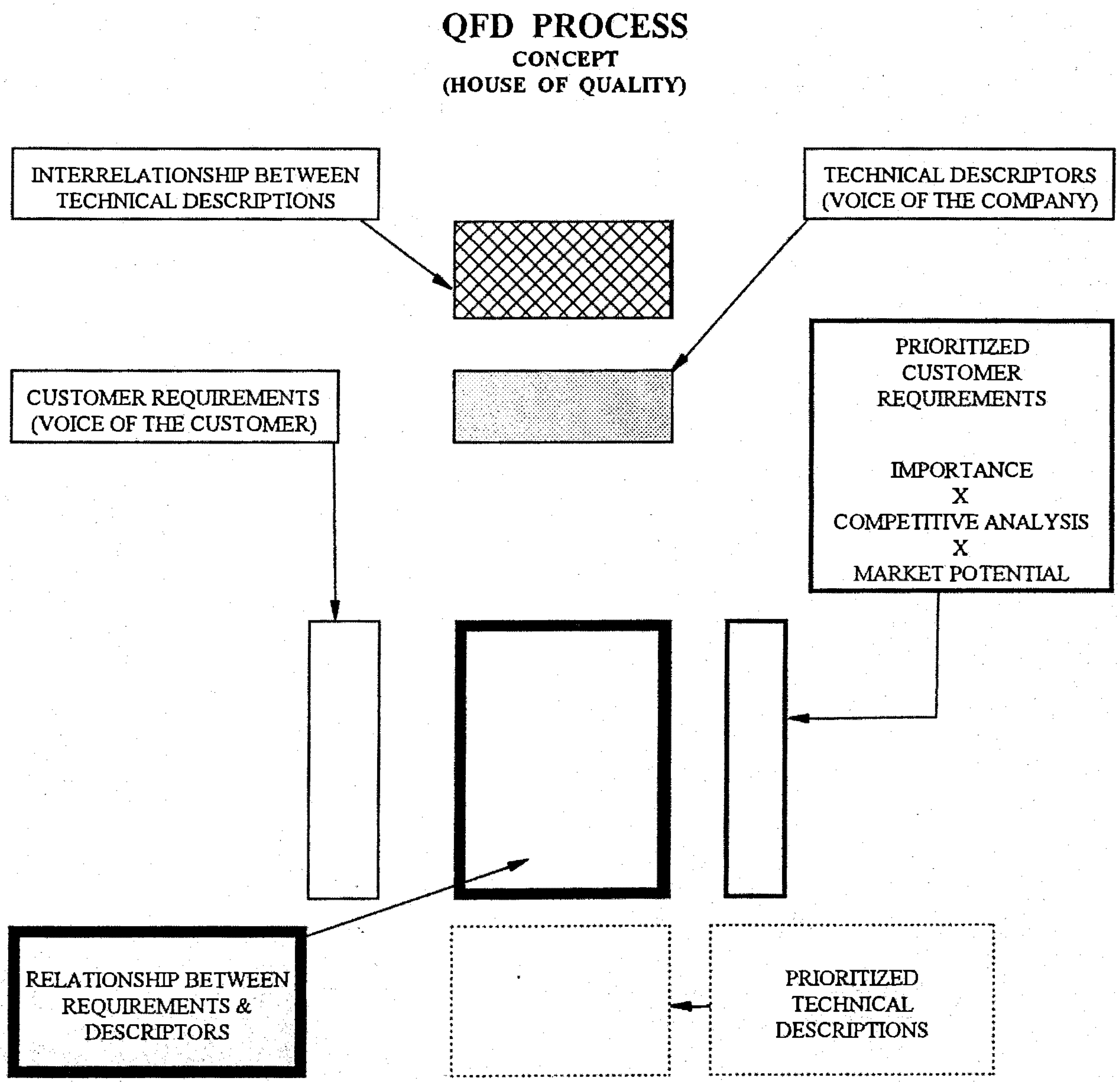

QFD PROCESS
CONCEPT
(HOUSE OF QUALITY)
INTERRELATIONSHIP BETWEEN TECHNICAL DESCRIPTIONS
TECHNICAL DESCRIPTORS (VOICE OF THE COMPANY)
CUSTOMER REQUIREMENTS (VOICE OF THE CUSTOMER)
PRIORITIZED CUSTOMER REQUIREMENTS
IMPORTANCE
X
COMPETITIVE ANALYSIS
X
MARKET POTENTIAL
RELATIONSHIP BETWEEN REQUIREMENTS & DESCRIPTORS
PRIORITIZED TECHNICAL DESCRIPTIONS

Figure 3.6 A Basic QFD matrix showing the various components

The technique was first introduced in Japan in 1972 and its first application was in shipyards (the Mitsubishi Kobe shipyard). Currently, the automotive industry is the biggest user. It has been successfully applied by companies such as Toyota, Ford and General Motors (Hauser & Clausing 1988).

Erikkson and McFadden (1993) regard QFD as the management approach that is used to facilitate company-wide quality control in practise. It can be applied in the planning, production and control processes. It is a technique that further encourage team work and the communication process between different departments within a company.

The technique is described in detail in Bossert (1991).

The application of quality function deployment as a tool to improve software quality is described in Erikkson and McFadden (1993). The QFD technique is used for the translation of customer requirements to specifications for the software and the associated metrics needed.

Erikkson and McFadden (1993) summarize the following positive aspects of using QFD in a software environment:

1) it brings the customer right into the design process and helps in prioritizing requirements for the product to be developed
2) it encourages defect prevention
3) it is a communication vehicle
4) important customer requirements can be traced to the related software characteristics, product features and product metrics
5) it gives the opportunity to follow the consequences process.

Erikkson and McFadden (1993) also note the following disadvantages of implementing quality function deployment in a software environment:

1) it requires an additional investment
2) it can be seen as reducing flexibility in the process and
3) can be difficult to administer, especially in the beginning stages of introducing the

technique.

Recently, Jacob, Luke and Reed (1995) have used quality function deployment to develop a process measurement program for software maintenance. The aim of the measurement program was to identify the critical metrics for the entire process.

3.4.5 BENCHMARKING

The main aim of benchmarking is process improvement

Douglas Cheney (Whiting 1991: 130)

Benchmarking can be defined as "the continuous process of measuring products, services, and practices against the company's toughest competitors and against companies regarded as industry leaders" (Fenwick 1991: 65).

Benchmarking has also evolved to describe a standard for comparison or a point of reference for other products or activities which are similar to the one which has been chosen to serve as the benchmark.

An excellent definition in the form of a menu (Spendolini 1992:10) is reprinted here:

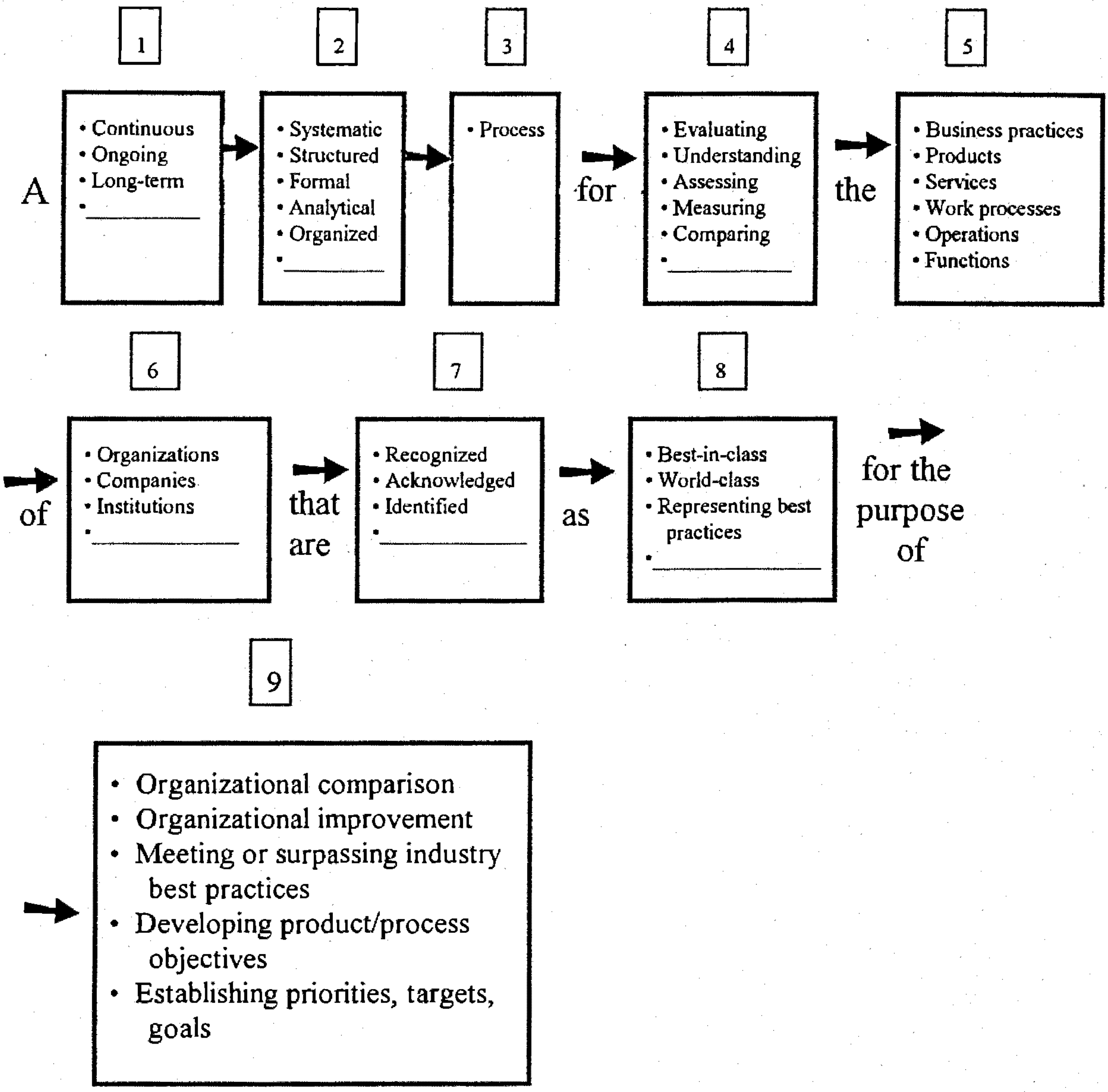


Figure 3.7 The benchmarking menu

Splendolini (1992) describe the benchmark process as a five-stage process:

1) determine what to benchmark

2) the forming of a benchmark team. Benchmarking is done by teams to take advantage of the diversity of knowledge, skills and perspectives that groups offer, as well as to balance workload and time requirements (Spendolini 1993: 53).

3) The identification of benchmark partners
4) The collection and analysis of the benchmarking information
5) Act on the information

Karlöf and Östblom (1993) have a slightly different break-down of the process. They do not identify the forming of a benchmark team as a stage on its own and consider stage four of Spendolini (1992) as two stages. Their five stages are:

1) decide what to benchmark
2) the identification of benchmarking partners
3) the gathering of information
4) the analysis of the information
5) the implementation of the results.

Interested readers are referred to the books: *The Benchmarking Book* by Michael J. Spendolini (1992) and *BENCHMARKING: A signpost to excellence in quality and productivity* by Bengt Karlöf and Svante Östblom (1993).

3.4.6 THE SEVEN PLANNING TOOLS

Deming, in 1950, used the following diagram (figure 3.8) to illustrate the steps we need to use in managing a business. The effective use of implementing this cycle in business was, however, limited.

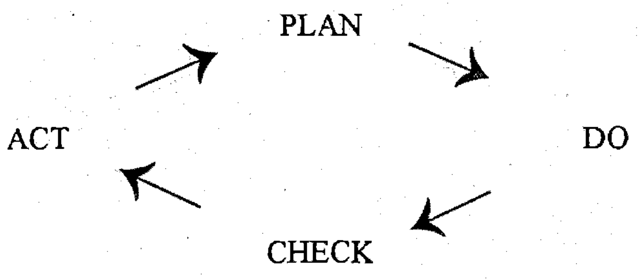


Figure 3.8 The Plan-Do-Check-Act Cycle

The seven planning tools provide managers with the tools needed for effective planning. The tools also provide individuals with the ability to contribute to the planning steps.

A summary of the tools follows (Bossert 1991: 48-50):

1) ***The Affinity Diagram (KJ Method)***

This tool gathers large amounts of language data (ideas, opinions, issues, etc.) and organizes it into groupings based on the natural relationship between each item. It is largely a creative rather than a logical process.

2) ***The Interrelationship Digraph***

This tool takes complex, multivariable problems on desired outcomes and explores and displays all of the interrelated factors involved. It shows graphically the logical (and often causal) relationships between factors.

3) ***Tree Diagram/System Flow Diagram***

This tool, which resembles a horizontal organization chart, systematically maps out the full range of tasks/methods needed to achieve every GOAL/purpose. The very structured process translates the most general goal into the practical implementation steps that need to occur.

4) ***Matrix Diagram***

This tool takes the necessary tasks (often from the Tree Diagram) and graphically displays their relationship with people/functions or other tasks. This is frequently used to determine who has responsibility for the different parts of an implementation plan.

5) ***Matrix Data Analysis***

This is the most statistically sophisticated of the New Tools. Its graph shows the strength of the relationship between variables which have been statistically determined. This is frequently used in marketing and product research.

6) ***Process Decision Program Chart (PDPC)***

This tool maps out every conceivable event and contingency that can occur when moving from a problem statement to the possible solutions. This is used to plan each possible chain of events that need to happen when the problem or goal is an unfamiliar one.

7) ***Arrow Diagram***

This tool is used to plan the most appropriate schedule for any task and to control it effectively during its progress. This is closely related to the CPM and PERT Diagram methods. This is used when the task at hand is a familiar one with subtasks that are of a known duration.

These tools are comprehensively described in Bossert (1991).

3.4.7 THE ISO 9000 SERIES OF STANDARDS

The International Organisation for Standardisation develop an international quality system standard in an effort to eliminate global confusion and conflicts about standards. The ISO 9000 series was issued in 1987. It exists of five parts: ISO 9000, 9001, 9002, 9003 and 9004. ISO 9000 and ISO 9004 are guidelines and ISO 9001, 9002 and 9003 are categories for which companies can apply for certification (Arnold 1994).

The underlying philosophy of this series of standards is that businesses must address specific elements (e.g. design control; purchasing; statistical techniques) in order to be successful. The International Organisation for Standardisation approach is that ISO 9001, 9002 and 9003 are minimum guidelines that should be followed and not be the only result to work towards. It was written in such a way that it can be successfully implemented in almost any type of business without modification or change.

The ISO 9001 is the most well known and is described as: *ISO 9001 QUALITY SYSTEMS - MODEL FOR QUALITY ASSURANCE IN DESIGN, DEVELOPMENT, PRODUCTION, INSTALLATION AND SERVICING* - This standard describes the quality system used to support the development of a product which involves design.

Of interest to the software industry is ISO 9000-3 and ISO 9004-2. ISO 9000-3 contains guidelines for the application of ISO 9001 to the development, supply and maintenance of software and ISO 9004-2 provides guidelines for the servicing of software facilities such as user support (Arnold 1994).

The requirements of the standard are partitioned into 20 headings. A summary of each is included (Thornton 1994: 2-19 - 2-21):

Management Responsibility

The model emphasizes the importance of management in quality control throughout the organisation. The clause sets out the basic principles for establishing the quality system within the organisation and sets out many of its functions, which are described in detail in later sections.

Quality System

The model requires the organisation to set up a quality system. The focus of the plan should be to ensure that activities are carried out systematically and that they are well documented.

Contract Review

This specifies that each custumer order should be regarded as a contract. Customer requirements should be clearly defined and in writing. Differences between the order and the original quotation should be highlighted. It should be ensured that the requirements can, in fact, be met.

Design Control

Design control procedures are required to control and verify design activities, to take the results from market research through to practical designs.

Document Control

Three levels of documentation are recognised by the standard.

Purchasing

The purchasing system is designed to ensure that all purchased products and services conform to the requirements and standards of the organisation. The emphasis should be placed on verifying the supplier's own quality management procedures.

Purchaser supplied product

All services and products supplied by the customer must be checked for suitability.

Product identification and traceability

Procedures must be established to identify and trace materials from input to output.

Process Control

This must be documented and procedures for setting up or calibration must also be recorded.

Inspection and Testing

This is required to ensure conformance on incoming materials and services, 'in process' to ensure that all is going according to plan, and on the finished product or service.

Inspection, measuring and testing equipment

Any equipment used for measuring and testing must be calibrated and maintained.

Inspection and testing status

Materials and services are either awaiting inspection or testing, or they have either passed or failed inspection. This status should be clearly identifiable at any stage.

Control of non-conforming product

Although this clause is not prescriptive about performance levels, all non-conforming products or services need to be clearly identified and documented. Procedures to handle these products should be established.

Corrective action

Corrective action should be implemented via a systematic programme and records should be *kept of any action taken.*

Handling, storage, packaging and delivery

This clause covers all activities which are the contractual obligation of the supplier with regard to the handling of the product.

Quality records

These form the basis for quality audits. Existing practice should be assimilated wherever possible in order to reduce rework in the reproduction of previously established quality records.

Internal quality audits

The quality system should be inspected from within the organisation according to established procedures. Internal audits should be carried out in order to identify problems early on in the development cycle.

Training

Written procedures should be produced in order to establish training needs, carry out effective training and to record the training requirements and completed activities of all personnel.

Service

Documented procedures should exist to ensure that servicing is actually carried out and that there are sufficient resources available to provide this facility.

Statistical Techniques

The standard does not specify particular techniques or methods but says that those used should be appropriate for the intended purpose.

The process of becoming ISO 9001 accredited will differ from country to country. In South Africa, the SABS (South African Bureau of Standards) is used as accreditation body.

3.5 CONCLUSION

Measurement and information is a key component of the TQM infrastructure.

The strategic importance of quality was only recognised and embraced since the 1970's. In the 1990's, measurement and analysis are the instruments through which quality is managed strategically.

Supplier measurement; customer measurement; performance measurement; internal process measurement and the link of these measures to each other and to financial measures by means of quality information systems, within an organisation, has become vital for competetence.

The measurement of quality management in organisations is an additional dimension of the measurement of quality at the strategic level. Saraph et al (1989), Benson et al. (1991) and Flynn et al. (1994) have provided a basis for research into this aspect which should be further pursued.

The innovative quantitative structures and techniques, discussed in 3.4, currently used in industry as a vehicle to support, control and measure quality improvement, have been applied successfully in business. Case studies on companies that have applied these techniques provided valuable knowledge and, almost always, render aspects that need further research.

4 SOFTWARE METRICS PROGRAMS

"Would you tell me, please, which way I ought to go from here?"
"That depends a good deal on where you want to get to", said the Cat.
" I don't much care where ____ " said Alice.
"Then it doesn't matter which way you go", said the Cat.
" ------ so long I get somewhere", Alice added as an explanation.
"Oh, you're sure to do that", said the Cat, "if you only walk long enough".
Alice in Wonderland (Lewis Carroll 1865)

Once again, Alice's words depict businesses today that have no clear goals or mission. Only, the *somewhere* can mean the liquidation of the business. Knowing where you are (by means of measurement) and a vision of where you want to be (in terms of measurable goals) are essential in today's competitive environment.

4.1 INTRODUCTION

As software has become a major role player in today's business, improved software development is critical for the software industry. Improved software development entails a priority for improved software product quality and performance and development team productivity. Möller and Paulish (1993: 1) remark: *While computer hardware performance has been doubling approximately every three years, improvements in software productivity have been increasing at a modest 4% annual rate (Jones, 1991; Putnam, 1991).*

The three dimensions of software production: cost, quality and time need to be measured in practice. If one dimension is ignored, problems will occur in that dimension. Recent studies indicate that less than *1% of completed large software systems are typically finished on-time, within budget, and meet all user requirements* (Möller & Paulish 1993: 1).

The aim of developing software that is on time, within budget and of good quality has led many software organisations to adopt a software metrics program in order to help them to measure all aspects of the development process as well as other key factors that influence their businesses' performance. Software metrics are considered essential to be able to understand, control and manage the software development process (Pfleeger 1991).

Most executives believe that quality and productivity are the most critical issues that face their organisations but do not know how to achieve it (Shetty, reported in I/S Analyzer 1994: 18). A software metrics program is a part of the continuous quality improvement process.

The role of a software metrics program, by measuring variables in each of the key areas that impact software development in the organisation in question, is to identify *strengths and weaknesses, pinpoints areas for improvement, makes recommendations and provides follow-up measures to show progress over time.* A quantitative and qualitative baseline, i.e. a benchmark of key factors impacting the organisation is established in this way (Case Study: Hewlett Packard, in I/S Analyzer 1994: 2). A software metrics program is also positively viewed by the customer who realises that a company is taking steps to improve its weaknesses (Möller & Paulish 1993). It is also instrumental in understanding and managing risk (Stark & Durst 1994).

Software metrics programs are not magic wands. Grady and Caswell (1987) advise that it takes at least three years of planning, collecting and analysing data before sufficient data is available to detect trends for an entire organisation. It is also of basic importance not to regard the collection of software metrics as an independent goal. It can only be successful if it is part of the process of managing software development and represent a long-term management commitment to understanding and managing software development better (Grady & Caswell 1987). By viewing it as a tool for managing software development, its role becomes clear. The effectiveness of linking the metrics data to actions designed to improve the process, will determine the overall improvements achieved (Möller & Paulish 1993).

Awareness of the important role of measurement in software engineering is reflected in the explosion of this activity, both in research and commercial applications. Measurement is becoming an integral part of all software activity. The metrics philosophy: ***Measurement is not the goal***

The goal is improvement through measurement, analysis and feedback (Daskalantonakis 1992: 1010) summarizes the role of measurement in software very aptly.

As mentioned in chapters 2 and 3, it is important to improve the design and building of processes within an organization to improve product quality. Measurement of the processes will lead to better understanding and increased predictability of the processes. It provides visibility of the whole process.

This chapter will deal with the following managerial and organisational aspects regarding software metrics programs:

1) definitions of terms used
2) aspects of implementation such as:
    - organisational requirements
    - different measurement approaches
    - planning a practical implementation framework
    - a company-wide database
    - measurement tools
    - the human aspects
    - training
    - implementation problems
    - evaluation and feedback.

The state of software metrics program practices worldwide will then be described. The extension of the concept of the metric approach to other industries will be discussed.

4.2 DEFINITIONS

Software metrics terminology is not standardized. The term software metrics is used to describe, for example, the discipline, the characteristics that are measured, the units in which they are measured and the actual values themselves! (NCC Fact Sheet 1992).

An overview of definitions contained in the literature follows.

4.2.1 MEASUREMENT

Measurement is the process by which numbers or symbols are assigned to attributes of entities in the real world in such a way as to describe them according to clearly defined rules (Fenton 1991: 2).

4.2.2 ATTRIBUTES

The software characteristics that are measured will be referred to as *attributes of specified software objects* (NCC Fact Sheet 1992: 6).

4.2.3 A SOFTWARE MEASURE

A dimension, attribute, or amount of any aspect of a software product, process, or project (Hetzel 1993: 3).

4.2.4 SOFTWARE METRICS

This section portrays the diversity that exists in the definitions and classifications of **software metrics** used by various authors.

4.2.4.1 Definitions

1) ***The scales or units used to measure the attributes*** (NCC Fact Sheet 1992: 6).

2) ***A standard way of measuring some attribute of the software development process*** (Grady & Caswell 1987: 4).

3) ***A method of quantitatively determining the extent to which a software process, product, or project possesses a certain attribute*** (Daskalantonakis 1992: 998).

4) ***Measurements used to compare software processes and projects or predict software outcomes*** (Hetzel 1993: 5).

Hetzel (1993: 5), however, also defines **software meters** as ***measurements used to control or regulate a software activity or process.***

The above distinction between software metrics and software meters is not a standard definition, but Hetzel's own interpretation.

4.2.4.2 Classifications

A distinction is made between **primitive** metrics (directly measurable or countable) and **computed** metrics by Grady and Caswell (1987: 4).

Möller and Paulish (1993: 40-43) introduce two further classifications, namely

1) **Objective** metrics (metrics that are easily quantified and measured) as opposed to **subjective** metrics that *attempt to track less quantifiable data* such as e.g. attitudes of personnel towards the use of CASE tools.
2) **Global** metrics *(high-level indicators that may span multiple phases of the software development process)* and **phase** metrics *(metrics that are indicators only for a specific phase of the development process).*

Daskalantonakis (1992: 999) categorizes software metrics as **process** metrics, **product** metrics and **project** metrics which he defines as follows:

Process metrics are metrics that are used for *improving the software development and maintenance process.*

Product metrics are metrics that are used to improve the software product.

Project metrics are metrics that are used *for tracking and improving the project.*

Conte (1986: 24) distinguish between **obtrusive** and **non-obtrusive** measures.

Obtrusive measures *require the involvement of the subjects* (e.g. through interviews or forms) while **non-obtrusive** measures are *observations of the program development process that are transparent to the subject.*

It is important to be aware of the differences and similarities between the definitions given by different authors. Global metrics and process metrics, as defined above, will in many instances, indicate the same group of metrics. Phase metrics, however, can include both product and project metrics.

4.2.5 USEFULNESS OF METRICS

An additional aspect that is stressed by many authors (DeMarco 1982; Daskalantonakis 1992; Conte 1986; Hetzel 1993) is the usefulness of metrics. Daskalantonakis (1992: 999) lists the following characteristics of a useful metric. They must be:

1) simple to understand and precisely defined
2) objective
3) cost effective
4) informative

Hetzel (1993: 4) defines **useful software measures** as *those that support effective analysis and decision making and that can be obtained relatively easily.*

Conte (1986: 22) mentions the following aspects, calling it meta-metrics, that need to be addressed in an industrial or experimental situation:

> ***SIMPLICITY*** *- Does the metric lead to a simple result that is easily interpretable?*

VALIDITY - Does the metric measure what it purports to measure?
ROBUSTNESS- Is the metric sensitive to the artificial manipulation of some factors that do not affect the performance of the software?
PRESCRIPTIVENESS - Can the metric be used to guide the management of software development or maintenance?
ANALYZABILITY- Can the value of the metric be analyzed using standard statistical tools?

Pfleeger (1993) adds that a software metrics program will only succeed and be a welcome part of software development and maintenance in an organisation if the metrics chosen are clearly needed and relatively easy to understand.

4.2.6 CONCLUSION

Great care must thus be taken in defining metrics. Comparison between companies can only take place if the measures used are defined in exactly the same way and the companies produce the same type of software (NCC Fact Sheet 1992).

4.3 IMPLEMENTATION ASPECTS OF A SOFTWARE METRICS PROGRAM

4.3.1 ORGANISATIONAL REQUIREMENTS

Defined organisational requirements form the basis from which the software metrics program implementation strategy will be planned.

Rubin (reported in I/S Analyzer 1994) recommends that the first step that an organisation should take in putting a measurement program in place is to assess its “measurement readiness”. A quick assessment method, suggested by Ruben, is the following:

TEST YOUR SOFTWARE PRODUCTIVITY MEASUREMENT READINESS

Score: 30 and above - strong. 15 to 25 - average. Below 10 - low.

1) How intense is the organisation's desire to improve its performance?
From: 0 (no desire), to : 5 (intense).

2) Is the organisation willing to invest time and money to improve systems performance with measurement?
From: 0 (no), to: 5 (funds and people are allocated).

3) What is the current level of systems skills inventory in regard to being able to use metrics?
From: 0 (none), to: 5 (already in wide effective use).

4) To what extent are measurement concepts known and understood by the system staff?
From: 0 (no staff has been exposed), to: 5 (100% trained).

5) Is the systems culture adverse to using measurements at the organisational and individual level?
From: 0 (100 % against), to: 5 (anxious to implement).

6) To what extent is a support structure in place to foster measurement practices and perform metric technology transfer?
From: 0 (none in place), to: 5 (in place).

7) Are tools and repositories for acquiring and analysing metric data in place?
From: 0 (no), to: 5 (full suite available).

8) Does the systems organisation understands its role in the business process?
From 0 (no), to: 5 (yes, the business processes are documented and tracked through metrics).

It is difficult to propose and validate the necessary organisational requirements for the establishment of a successful metrics program. Recommendations regarding organisational requirements for a software metrics program fall into four perspectives: context, inputs, process and products. The following summary provides an integration of the factors for each perspective, cited in Jeffery and Berry (1993):

1 CONTEXT:

The environment in which the metrics program is develop and operated.

It is important to:

a) Have clearly stated objectives and goals.

b) Have realistic assessment of pay-back period.

c) Have senior management commitment and support.

d) Have a quality environment established as well as stable development processes.

e) Determine the required granularity.

2 INPUTS:

Factors or resources that are applied to the metrics program.

It is important to:

a) Resource the program and measurement team properly.

b) Allocate resources to training to motivate and sustain interest.

3 PROCESS:

The method used to develop, implement, and maintain the program.

It is important to:

a) Let the objectives determine the measure.

b) Have an independent metrics team.

c) Create a metrics database.

d) Use automatic tools where possible.

e) Use measures only for pre-defined objectives.

f) Let everybody knows what is being measured and why, and to develop and publish an implementation plan.

g) Clean and use the data promptly.

h) Make measurement active by integrating measurement and process.

i) Provide capabilities for users to explain events and phenomena associated with project.

j) Provide an extensible framework for the addition of new techniques.

4 PRODUCTS:

The measures taken, reports produced and other output of the program.

It is important to:

a) provide feedback on results

b) facilitate actions to be taken on basis of observed measurements.

Möller and Paulish (1993) emphasize the fact that the successful implementation of software metrics in an organisation is highly dependent on the level of support provided from top management, also mentioned in Jeffery and Berry (1993) and Fenton (1991). This will hold true, regardless of the measurement approach (see 4.3.2) chosen.

The aspect of the establishment of a software metrics team is especially important within the organisational context. Credibility of the people involved can make or break the program. Grady and Caswell (1987) describe the Software Metrics Council (to be discussed in 4.3.3.5) that has been established within Hewlett-Packard in 1983 to form a foundation from which metrics activities are planned and executed. Fenton (1991) mentions that a measurement program will only be taken seriously if the right people are given the responsibility for it. According to Fenton (1991), the appointment of senior people within a company on the team, sends a signal that top management take the measurement program seriously. He further emphasizes the fact that the members of the metrics team should be volunteers, have an understanding of the importance of measurement and be trained in all aspects concerning measurement. This aspect is further discussed in 4.3.3.5.

Grady (1992) mentions the extreme importance of convincing the people involved of the importance of measurement, and the subsequent development of an environment of trust with consistent, correct use of data. Fenton (1991) also emphasizes that a measurement programme will only succeed if it has the full support of all the relevant personnel. The crucial role of the personnel involved in the Software Metrics Program is discussed in 4.3.6.

Daskalantonakis (1992) lists the following dimensions that need consideration when implementing a metrics program in an organisation: metric usefulness/utility (see 4.2.5), metric types or categories (see 4.2.4.2), metric audiences and users and their needs (see 4.3.6), and the levels of metric application. Levels include the company (or business unit) level, the product group level, the project level and the component (e.g. subsystem of a project) level.

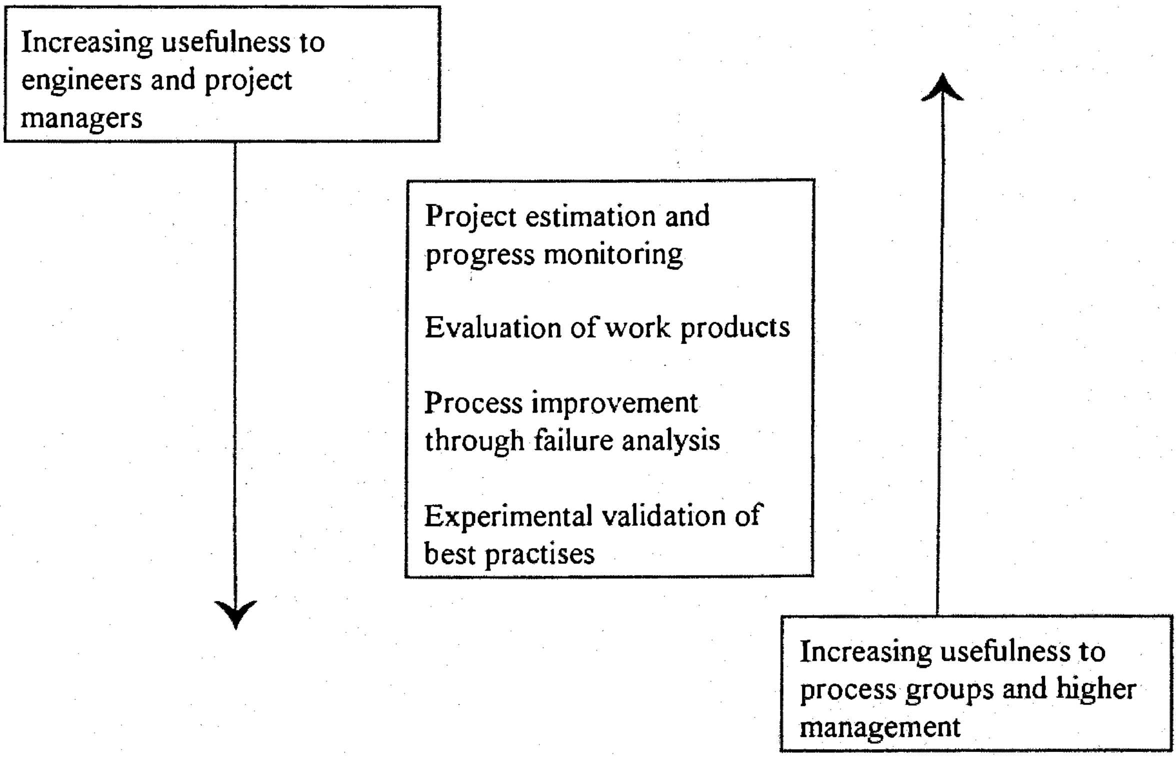
Daskalantonakis (1992) emphasizes the critical importance of a **software metrics infrastructure** in an organisation to facilitate the implementation of metrics. This consists of working groups with participation across the company, the deliverables (e.g. metric documentation), training workshops on metrics (to be discussed in 4.3.7), tools automating metrics (to be discussed in 4.3.5) and consulting support for metric implementation within projects (to be discussed in 4.3.7). He also mentions additional activities and outputs that are part of the software metrics infrastructure and that has been established by the Metrics Working Group in the Motorola company. They are:

1) clarifying metrics definition, interpretation and use. Metrics users in the company receive it through metrics documentation and training material.
2) support for further analysis of collected data through the use of generic defect classification schemes and examples on how to use these schemes to create process improvement recommendations
3) the use of the Defect Prevention Process (Jones 1991) as an effective tool to ensure process improvement through analysis of data on defects
4) the provision of guidelines to create a function responsible for implementing software metrics for business units
5) a method for assessing software measurement technology has been created
6) customer satisfaction measurement through surveys is encouraged

4.3.2 MEASUREMENT APPROACHES

Different measurement approaches can be followed when setting up a metrics program. Two major measurement approaches, top-down and bottom-up, are advocated in the literature. I have termed them Global (top-down) and Project-Oriented (bottom-up) Software Metrics Program Strategy approaches.

Grady (1994: 19) describes (depicted in figure 4.1) the major uses of software metrics and the conflicting pressures on data within an organisation. When deciding on the approach to be followed, these pressures need to be identified and taken into account.



Increasing usefulness to engineers and project managers

Project estimation and progress monitoring

Evaluation of work products

Process improvement through failure analysis

Experimental validation of best practises

Increasing usefulness to process groups and higher management

Figure 4.1 Major uses of software metrics

The two approaches will now be discussed.

1 THE GLOBAL APPROACH

This is a top-down strategic approach where one *starts with high-level goals and needs* *and derives the measures needed to support them* (Hetzel 1993: 26).

The Goal-Question-Metric (GQM) Paradigm (Basili & Weiss 1984) is the most commonly used instrument to establish the metrics to be used. The basic steps involved (Hetzel 1993: 27) are:

1) The development of clearly defined goals. (The goals can be corporate, divisional or project level. It usually addresses quality and productivity issues.)
2) The "generation of questions that define the goals. (A list of questions that need answers in order to know whether the goals have been met.)"
3) The identification of metrics that will answer the questions (the measures to be collected or tracked to answer the questions).

Grady and Caswell (1987: 17-18) suggest the following strategy, using a global approach, for implementing a metrics program in a company:

1. *Define company/project objectives for program*
   This will determine the methods to be used, costs of the program and the level of support from top management.
2. *Assign responsibility*
   Organisational location of responsibility for metrics and the people used to implement the program indicates the importance of the program.
3. *Do research*
   Literature research is needed in order to make decisions regarding the metrics to be implemented.
4. *Define initial metrics to collect*
   A simple set of a few important metrics, such as metrics for size, defects and effort are recommended.
5. *Sell the initial collection of these metrics*
   The success of a metrics program depend on accurate data. That can only be achieved through commitment of the people collecting the data. The importance of metrics must be clear to all personnel within a company.

6. *Get tools for automatic data collection and analysis*

   Tools reduce time expenditure and ensure accuracy and consistency if applied correctly. Metrics Tools will be described in 4.3.5.

7. *Establish a training class in metrics*

   Training is essential for understanding the underlying concepts, reasons for, and importance of, metric data collection. This will apply to all people who are involved with collecting metrics within the organisation. Training aspects will be discussed in more detail in 4.3.7.

8. *Publicize success stories and encourage exchange of ideas*

   This provide feedback to people and motivate people to continue with the program.

9. *Create a metrics database*

   A database is necessary in order to evaluate trends and effectiveness. The establishment of a company-wide metrics database will be discussed in 4.3.4.

10. *Establish a mechanism for changing the standard in an orderly way*

    The process and metrics will evolve and mature over time. Feedback is continually required to update metrics and the program structure.

Grady and Caswell's (1987) strategy thus entails a small set of initial metrics that would measure the following criteria:

1) Size
2) People/Time/Cost
3) Defects
4) Difficulty
5) Communications

The metrics they selected were:

1) NCSS (noncomment source statements) as a standard metric for size.

   Noncomment source statements include compiler directives, data declarations, and executable code. Each physical line of code is counted once. Each include file is counted *once. Print statements are lines of code* (Grady & Caswell 1987: 58).

2) The payroll month as the standard metric for cost. Percentage of over/under time was also recorded.

3) A defect is a problem or an error, anything in the output of the software process which would not exist if the process was perfect. They define a **defect** as ***a deviation from the product specification or an error in the specification if the error could have been detected and would have been corrected. If the error could not possibly have been detected, or it could have been detected and would not have been corrected, then it is an enhancement, not a defect. Defects do not include typographical or grammatical errors in the engineering documentation.*** (Grady & Caswell 1987:56) and recorded defects introduced, found and closed within the four stages of specifications, design, implementation and testing.

Metrics for difficulty and communications were defined, but made optional.

Forms that were used by Hewlett-Packard for the collection of the metrics in their organisation are published in Grady and Caswell (1987).

A similar strategy, based upon Grady and Caswell's (1987) experience and recommendations, is suggested by Möller and Paulish (1993: 29-47). Their seven steps are:

1) *The Software Development Process*

   It is important that the software development process is documented and understood before a Metrics Program is initiated. It serves as a baseline process which will be measured and incrementally improved.

2) *Goals*

   The goals for the Metrics Program need to be identified. They need to be in synergy with the business goals of the company. The objectives should also be reviewed for consistency with any existing corporate or organisational initiatives for quality improvement for general activities. The inclusion of personnel, by asking what they want from metrics and what they can contribute, is an important part of this process. This also leads to support for the Metrics Program.

3) *Responsibility*

The assignment of the responsibility within the organisation for the Metrics Program and the individual(s) to implement the Program will be an indicator of the importance of the Program to the overall organisation.

4) *Initial Research*

This requires the initial information needed to establish the Metrics Program. Goals and customer expectations need to be validated through internal customer surveys and/or assessments.

5) *Metrics Definition*

The initial set of metrics needs to be defined. The metrics definition and the way the data is going to be collected should be described within a written Metrics Plan.

6) *Sell*

The Metrics Program needs to be introduced and communicated in such a way that cooperation of personnel and visibility throughout the organisation is achieved.

7) *Feedback and Process Improvement*

Establish the feedback mechanisms so that improvement actions can be identified and implemented.

Möller and Paulish (1993) thus advise the use of a global approach when starting a Metrics Program. They recommend the use of a limited number of initial basic metrics. They add that these metrics need to be precisely defined and communicated to all people involved in the Metrics Program to ensure visibility of the program.

Möller and Paulish (1993) suggest five global metrics for measuring progress that are easy to collect. The metrics will be listed, but not discussed. They are:

1) Lines of code (to measure size).
2) System test faults (an indicator of product quality).
3) Customer Change Requests (an indicator of product quality during field use).
4) Schedule (a measure of process quality).
5) Productivity (a measure of process quality).

Daskalantonakis (1992: 1001) describes Motorola's company-wide software metrics initiative. They have started with a set of metrics that address the following attributes set out in their Quality Policy for Software Development:

1) delivered defects and delivered defects per size
2) total effectiveness throughout the process
3) adherence to schedule
4) estimation accuracy
5) number of open customer problems
6) time that problems remain open
7) cost of nonconformance
8) software reliability.

A minimum set of attributes that need to be measured, suggested in the NCC Fact Sheet (1992) is:

1) Staff effort for development and maintenance.
2) Other costs (Training, tools, travels, etc.).
3) Project duration.
4) Post release defects and their origin.
5) Rework costs: pre- and post release.
6) Characteristics of product in its operational environment (e.g. time between failures, time to correct defects).

According to Clapp (1993), most organisations start with a simple set of metrics. She states that the most frequently used attributes that are measured in a software development company are size, personnel, computer use, unit progress, and problem reports.

Different sets of global metrics for initial data collection are thus defined in the literature. A global approach advocates the use of a few simple, but important and practical, easily understood, metrics to start with. Once these are established, metrics can be extended to measure all parts of the processes and products in the organisations.

2 THE PROJECT-ORIENTED APPROACH

The project-oriented approach is a bottom-up tactical approach (Hetzel 1993). It defines the set of required measurements at the engineering level and builds up to the management level.

Hetzel (1993: 29) suggests a *bottom-up measurement paradigm that specifies a base set of measurements to be collected on every software work product developed and used.* Hetzel (1993) argues that the principle behind a bottom-up approach is that measurement's primary role is to support the engineering activity.

Hetzel (1993) criticises the Goal-Question-Metric approach, discussed above, with regard to the aspect of goals. He argues that nobody in a company knows what the right set of goals should be and need good measurement to set their goals. He reverses Basili's GQM paradigm and suggests a MQG spiral (Hetzel 1993: 31).

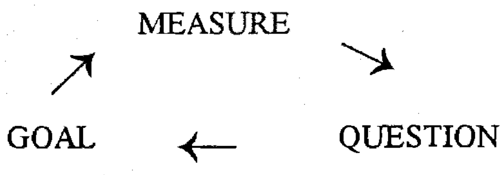


Figure 4.2 The MQG spiral

Hetzel (1993) emphasizes that measurement should come first, not last. Measurement helps in providing knowledge and insight about the engineering activities and will stimulate questions. The knowledge gained will result in setting goals and targets and to improve or change the process.

Furthermore, as this process is part of the software engineering activities, more involvement and support is gained from the people using it. The support of the personnel involved is a key element in the success of a software metrics program. They have to know that the measures are their to serve, and is not a direct performance instrument. Grady and Caswell (1987) also emphasize that widespread involvement of people using meaningful measures is necessary. The people closest to the process are the ones who can most quickly help to bring it under control, and measurements will help them to identify how. Pfleeger (1993) mentions the metrics team at the Contel Technology Center who felt that the people involved would only collect and analyze metrics correctly when the metrics meet a specific need or answer an important question. Lack of support from the people that need to implement the metrics as well as the tendency to "manipulate" the measured data are problems that can lead to the failure of the software metrics program.

Hetzel's measurement engineering bottom-up IOR (Input-Output-Results) model is depicted in figure 4.3 (Hetzel 1993: 32).

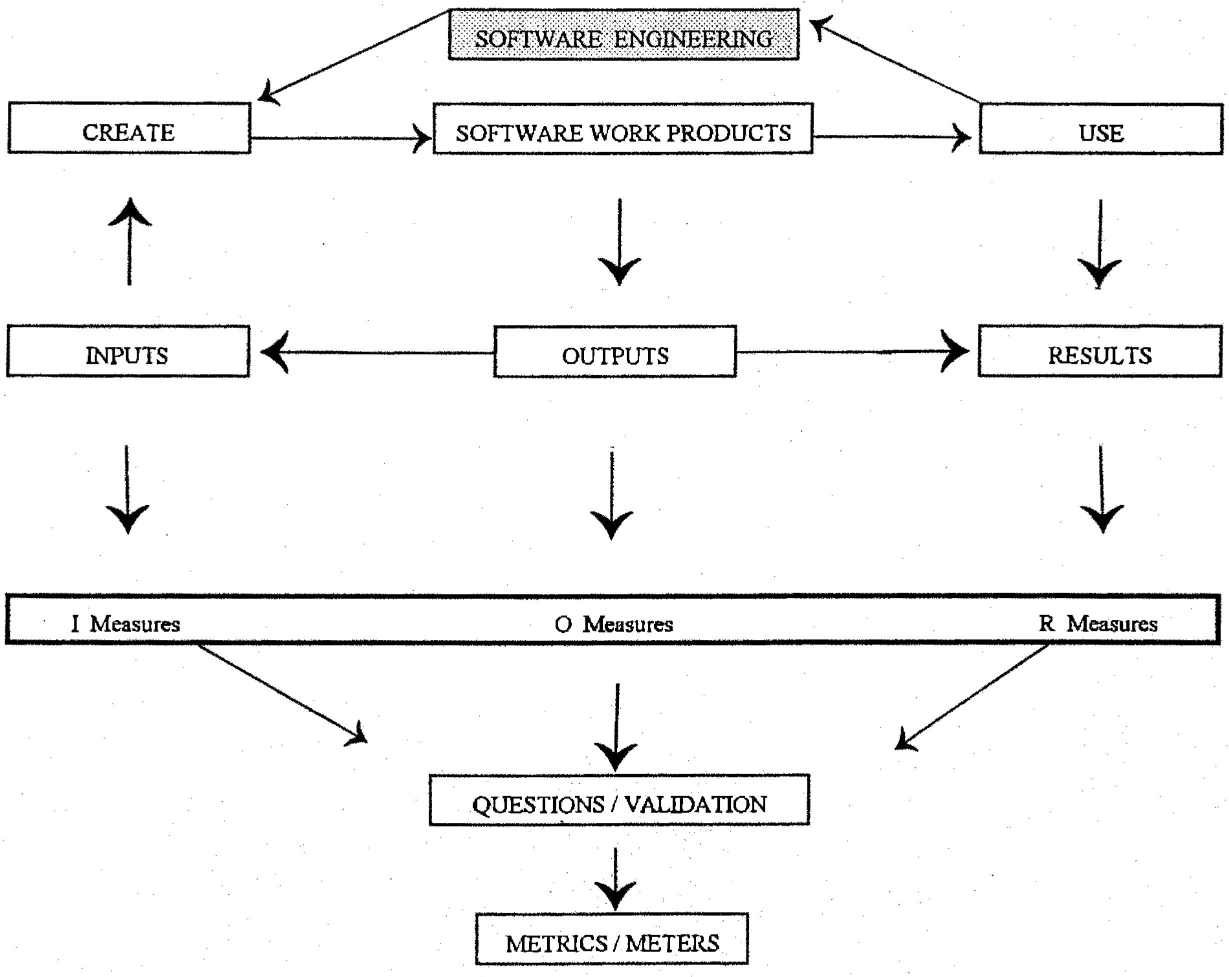


Figure 4.3 Measurement engineering bottom-up IOR model

The software work product measurements are defined as follows (Hetzel 1993: 29):

1 *INPUT MEASURES*

Information about the resources (people, computers, tools, other work products, etc.) applied and the process steps or activities carried out.

2 *OUTPUT MEASURES*

Information about the deliverables and work products that are created (e.g. size and complexity).

3 *RESULTS MEASURES*

Information about the usage and effectiveness (perceived and actual) of the deliverables and work products in fulfilling their requirements.

Heterogeneous projects (heterogeneous regarding processes, language, environment, tools, team structure and possible some other variables) exist in a number of organisations. A standard set of metrics (global approach) is not always advisable in such instances.

Pfleeger (1993) discusses her experience at the Contel Technology Center where it was decided that a standard set of metrics was inappropriate because of the heterogeneous nature of their projects. The team (part of the 13-member software engineering laboratory) has selected metrics, together with project personnel, with the Goal-Question-Metric paradigm according to each project's process maturity level. The Software Process Maturity Framework was discussed in chapter 3.4.1.

The five maturity levels (described in the context of the processes of a project) and the type of metrics that is suggested to be used at each level (Pfleeger 1993: 68) are:

Level 1: The process is not well defined and requirements are poorly understood. Measurement of effort and duration are suggested. This will provide a baseline against which improvements can be measured.

Level 2: The process is not fully understood but requirements are defined and structured. Project-management metrics that can establish general productivity measures are suggested.

Level 3: The process is clearly defined. Individual process activities are visible. Product measurement and the use of intermediate product characteristics to predict the quality of the final product is suggested.

Level 4: A project has a central point of control. Process measures with feedback to the responsible person is suggested. This information is used to make decisions about proceeding at critical points.

Level 5: It is suggested that a project at this level uses feedback and process measures to change the process dynamically. "Measurement guides change and control of processes."

The project-oriented measurement strategy followed by Pfleeger (1993) can be described as follows:

1) Select a few pilot projects so as to represent the different development types within the organisation.
2) Present a half-day workshop on metrics and cost estimation to the different project teams separately. This is followed by a half-day discussion of the status, requirements and problems of the project.
   A metrics plan that will address the specific needs of the project can then be drawn up.
3) The metrics team (one part-time and two full-time scientists in the instance of the Contel Technology Center) must monitor the use of metrics in the pilot projects. Pfleeger (1993) mentioned a workshop that was held for all the software managers after a few months at the Contel Technology Center. It was ended with a panel discussion where the experiences with the metrics were described. This workshop added to new interest in metrics.
4) The next phase is the provision of tools to support data collection and analysis at the project level. Metric tools are discussed in 4.3.5.

The approach thus favoured by Pfleeger (1993) and Hetzel (1993) emphasizes the important role of the person collecting and using metrics in the success of the metrics program.

Pfleeger (1993: 74) concludes: ... *the focus should be on solving project and process problems first, with institutional or organisational problems to be addressed later.*

DeMarco (1982), when discussing the use of metrics for project forecasting, also favours a project-oriented approach. He advises to start with two or more relatively new projects that are likely to be completed within a year, with a staff component of approximately 15. He adds that the initial period of uncertainty and poor data collection can be used to collect global information about the organisation.

Stark and Durst (1994) describe the metrics initiative at NASA's (National Aeronautics and Space Administration of the USA) Mission Operations Directorate (MOD) where the key requirement was the unobtrusive monitoring of a project's progress. Criteria for the selection of metrics that were used were:

1) metrics need to be relevant to the MOD development and maintenance environment
2) collection and analysis have to be cost-effective
3) multiple metrics are required during each reporting period for cross-checking and to provide a full picture
4) metrics need to have a strong basis in industry or government for *establishing 'rule of thumb' thresholds for use by project managers.*

They started with six projects over two years as a testing period. The initial data would also provide information for subsequent training. They implemented their initiative by means of a three step process. The three steps are definition, documentation and education. They applied Basili's Goal-Question-Metric paradigm for the definition step. Their documentation exists of handbooks that contains precise definitions and implementation details for managers and engineers as well as a metric toolkit. The toolkit will be described in 4.3.5.

3 A SUMMARY OF THE TWO APPROACHES

Table 4.1 indicates the two approaches, the procedure within each approach as well as the important advantages and disadvantages of each.

GLOBAL	PROJECT-ORIENTED
PROCEDURE: 1. Define the objectives of the metrics program 2. Assign responsibility for metrics program 3. Do a literature search and define the initial metrics to collect 4. Collect data initially by means of a form (manually or electronic) 5. Analyse the data 6. Provide feedback and discussion for improvement 7. Establish training in metrics 8. Automate metric data collection as far as possible	**PROCEDURE:** 1. Select starting projects so as to represent development types 2. Present workshops on metrics and cost estimation (½ day) and discuss project status, requirements, problems (½ day) 3. Select metric set according to "maturity" of each project by means of the GQM paradigm 4. Manual collection of data initially but simultaneously, develop a metric tool kit for selection by project managers
ADVANTAGES: Obtain an overall picture Fairly easy to collect Cost outlay minimal	**ADVANTAGES:** Gains direct and value-added for project and company The fact that the need for metrics will be seen and advocated by development people themselves is a successful strategy for the growth of a metrics program
DISADVANTAGES: Not value-added if projects are heterogeneous Not addressing immediate measurement needs as seen from project management side Gain minimal compared to time consumed from project management side	**DISADVANTAGES:** Cost outlay Time aspect for initial implementation (up to the level of the toolkit) for personnel involved

Table 4.1 Summary of the two measurement approaches

4.3.3 PLANNING A PRACTICAL IMPLEMENTATION FRAMEWORK

No program or initiative begins without the idea being proposed by a company member or consultant. It usually follows from the attendance of a conference/seminar and/or the reading of trade and academic journals. It could also be as a result of dealing with problem areas in the organisation.

Software metrics programs are, as mentioned before, a part of the continuous total quality improvement program. As such, they are usually included with the Quality Improvement Plan proposal for the organisation. The program needs support from top management and resources in order to be implemented successfully.

The following is a proposed practical framework to start off the process of metric collection that can be used with each of the above mentioned measurement approaches.

4.3.3.1 *Why* do we need measurements?

The first step is to introduce the concept of measurement to the personnel that will be involved in the metric collection and analysis process. This can be done by means of a presentation/workshop. This aspect is critical as it will determine the initial attitude towards metrics.

The presentation should entail a thorough, but short explanation, on the strategic and critical importance of measurement. Aspects such as the tracking and measuring of processes and products in the quest to develop cost-effective, quality and on-time software products as well as providing a communication vehicle between management and software product development personnel need to be stressed. It is important to "sell" the concept of metrics at this stage by means of examples from industry.

4.3.3.2 *What* are we going to measure?

The second step is to decide upon a measurement approach strategy (discussed in 4.3.2). Once this has been achieved, the initial set of metrics to be used, needs to be determined.

In deciding upon an measurement approach strategy, the company involved can use the following structure (adapted from Möller & Paulish 1993: 6) depicted in figure 4.4:

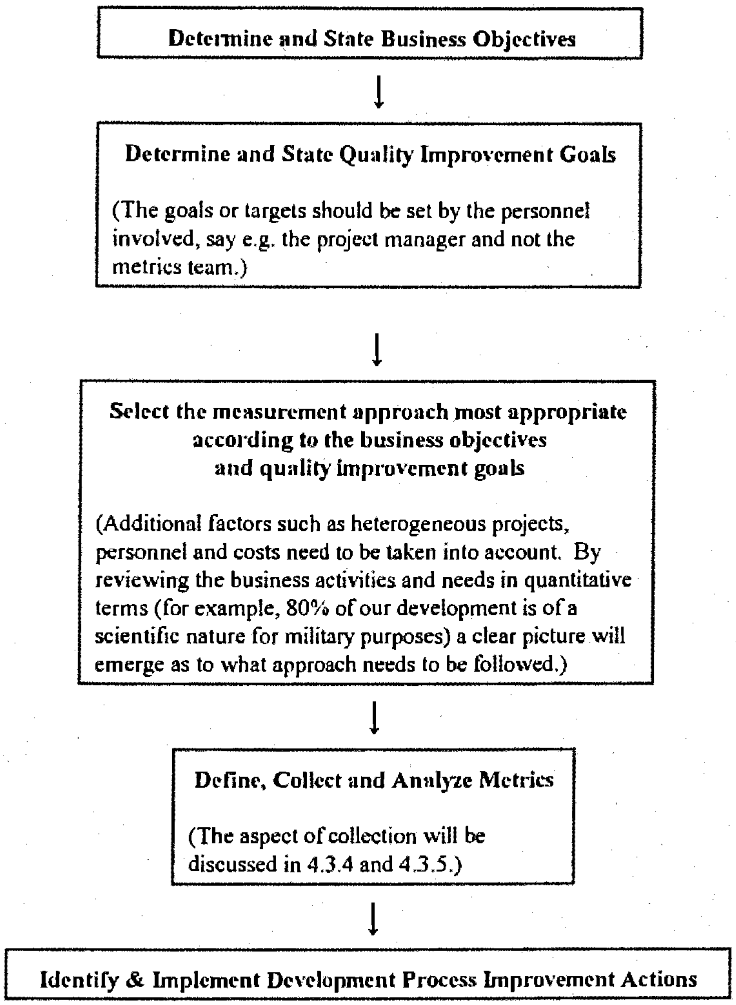

Determine and State Business Objectives
Determine and State Quality Improvement Goals
(The goals or targets should be set by the personnel involved, say e.g. the project manager and not the metrics team.)
Select the measurement approach most appropriate according to the business objectives and quality improvement goals
(Additional factors such as heterogeneous projects, personnel and costs need to be taken into account. By reviewing the business activities and needs in quantitative terms (for example, 80% of our development is of a scientific nature for military purposes) a clear picture will emerge as to what approach needs to be followed.)
Define, Collect and Analyze Metrics
(The aspect of collection will be discussed in 4.3.4 and 4.3.5.)
Identify & Implement Development Process Improvement Actions

Figure 4.4 A software metrics approach

Fenton (1991: 112) illustrates the chain from measurement to action as follows:

measurement → facts → decisions → action.

The Goal-Question-Metric paradigm can then be used as an instrument to determine the initial set of metrics to be used (Basili & Weiss 1984).

The personnel involved in the decision regarding the measurement approach need the approval of both top management and software engineers that are going to use the measures. To ascertain this, meetings between the team responsible for metrics and the parties concerned is important.

The attributes that need to be measured and the metrics that will be used in measuring these attributes are, as discussed, determined by the outcome on the approach chosen. Examples have been given when the approaches were discussed in 4.3.2.

4.3.3.3 *How* are we going to measure it?

This will be dependent on the decisions taken in step 2. Different options exist. It can be done manually, by means of a form, or by using an automated tool. Although the second option is preferable in terms of time and costs, the first option is usually viable and preferred as a starting point.

Examples of forms that have been published in Grady and Caswell (1987) as well as forms that have been developed for a software development company are provided in Appendix C.

The data collection process and the establishment of a company-wide database are discussed in 4.3.4.

Automated Metric Tools are currently encompassing such a wide variety that it is discussed under a separate heading (4.3.5).

An important aspect is the ultimate integration of this information with the management information system once the program is in progress as to ensure that management have access to critical information at the right time.

4.3.3.4 *When* are we going to measure it?

Time constraints will be discussed with respect to

1) the measurement approach strategy and
2) the specific measures .

1 MEASUREMENT APPROACH TIME CONSTRAINTS

Global: Time constraints will include the following: time spent on establishing the Metrics team and time spent by this team to reach consensus regarding the initial set of metrics to be used.

Data collection can start once the metrics and the format in which they will be collected have been defined. The format can be a form which needs to be completed or an automated version, e.g. input into a spreadsheet program. The establishment of the format also constitutes a time component.

Project-engineering oriented: The time constraints will include the identification of pilot projects and the time involved in organising and preparing for a workshop on metrics as suggested in Pfleeger (1993).

Toolkit development is done once the initial foundation is laid. One needs to keep in mind that this is a comprehensive exercise and involves several months. It is also a continuing process as information needs to be updated from time to time.

2 TIME ASPECTS OF METRIC DATA COLLECTION

The establishment of the time intervals and time units for the measurement of the different selected metrics is a critical point. It can lead to meaningless data if the data is not collected correctly time wise.

An example of this can be effort: it will be meaningless and impractical if software development effort is measured in minutes. The most practical time unit is the concept of a man-month.

Time units, like the man-month need to be defined clearly. It will ensure that personnel involved all use the definition stated, and not his/her own concept of a man-month. The same concept can be defined differently in different countries. A European staff-year differ from the United States definition of a staff-year (Möller & Paulish 1993). A rule to account for paid and unpaid overtime (and how this is included in the measurement of effort) is also necessary for correct interpretation.

Regarding time, extreme caution needs to be exercised if metrics tools are chosen and used. The tool's time definition may differ from what may be practical or from the only way you can measure a certain metric in your environment.

Another important aspect is the intervals between measurements. It is important to measure at the start of a project and then, say, monthly for global indicators. After each phase of development, actual and planned values also need to be compared.

4.3.3.5 *Who?*

This aspect concerns the people responsible for implementing the metrics program. This aspect has been addressed in terms of credibility and responsibility.

It is advised in the literature to establish an independent team of motivated people (Fenton 1991), at least of size three, to initiate and start the program (DeMarco 1982). DeMarco further advises that the personnel involved, should be assigned only half-time, and that the other half of their time be spend on something entirely different. He also advises that the team should report to someone outside the project(s) to be measured.

Responsibilities of the team include decisions, on data to be collected and tools to support the implementation, after consultation with the personnel involved. Validation of the data is another important responsibility (Fenton 1991). In a company-wide program, the metric team will also be responsible for enabling feedback, reviewing and changing company standards (Fenton 1991).

Grady and Caswell (1987) and Daskalantonakis (1992) describe their experiences with this aspect in the Hewlett-Packard (HP) and Motorola company respectively. A short description of their respective experiences follows:

1 The Hewlett-Packard Software Metric Council

Grady and Caswell (1987) describe the creation of the HP Software Metrics Council in August 1983. It consisted of an invited group of twenty software managers and developers from thirteen divisions. They were chosen on the grounds of their *software development experience, software management experience, interest, prior work in software measurement, and/or influence within their organisational entities to implement the council's decisions. Personal commitment and enthusiasm were also important. In addition, developers of all the various types of HP software were represented* (Grady & Caswell 1987: 45).

Common terminology and measures for the process of software development that could be used throughout HP, early enough in the development process to effect change, was needed and had to be addressed by the Council.

The objective of their first meeting was: *To gain agreement on a set of software measurement criteria which managers feel are meaningful, reasonable to collect, and can be used to measure progress and predict results* (Grady & Caswell 1987: 45).

The meeting was held away from the offices and consisted of an industry report (a presentation of a literature survey), a guest presentation by Barry Boehm (author of the famous **Software Engineering Economics**) and reports regarding data collected and analyzed currently in the different divisions of HP. It helped in creating a common base of understanding regarding metrics. Workshops were the focus of the meeting. Consensus on criteria and the metrics to measure these criteria was achieved and is described by Grady and Caswell (1987) as a key step in establishing a metrics program throughout the company.

The responsibilities of the council members include: presentations to engineers, project managers and division management; consulting with team members from projects who wanted help in collecting and analysing data; and collecting feedback on meaningfulness and ease of use of the proposed metrics after a six-month period.

2 **The Metrics Working Group in Motorola**

Daskalantonakis (1992) describes the Metrics Working Group (MWG) that was established in Motorola, with participation from all the business units. Its aim was to define a minimum set of software metrics to be used company-wide for measuring and eventually improving the quality of the software.

The Group worked for three years intensively to define a common set of metrics. It also supports the process of implementing software metrics within the software development groups. Daskalantonakis (1992) mentions that debate lasted for about a year on the set of common metrics. It was then decided to rather start of with a set of metrics that addressed the measurement/improvement areas identified, and improve these metrics over time, instead of debating forever, trying to find the perfect set of metrics.

Motorola has also established a Metrics User Group (MUG). This group has reptesentation across business units and meets four times a year. They share experiences regarding tools, including demonstrations of tools and implementing metrics in projects. They are also involved in organizing an Annual Software Metrics Symposium within the company.

4.3.3.6 *So what?*

The last step, which also acts as a feedback instrument, is the evaluation of results in terms of quality and cost-benefits. This includes analysis of the collected data, reports of problems encountered with metrics and their collection, and modifications to definitions and procedures if required.

CONCLUSION

By addressing the **why, what, how, when, who** and **so what** aspects a clear picture will emerge on the organisation's structure for starting a metrics program. The framework is an effective instrument to brainstorm and develop a Software Metrics Program Plan.

4.3.4 A COMPANY-WIDE DATABASE

4.3.4.1 Data collection

Data should be collected with a clear purpose in mind. Not only a clear purpose but a clear idea as to the precise way in which they will be analysed so as to yield the desired information....It is astonishing that men, who in other respects are clear-sighted, will collect absolute hotch-potches of data in the blithe and uncritical belief that analysis can get something out of it.

Facts from Figures (M.J. Moroney 1950).

Data collection is the most critical part of the software metrics program. Without accurate, on-time and sufficient data no software metrics program can succeed. Data collection provides the direct measurements on which all subsequent analysis are based. Mellor in Fenton (1991: 89) remarks: *each item of data must contribute to a direct measure, on a meaningful scale, of some attribute of the processes, products, or resources with which we are concerned.* The role of data collection in measurement is depicted in the figure 4.5 (Fenton 1991: 90):

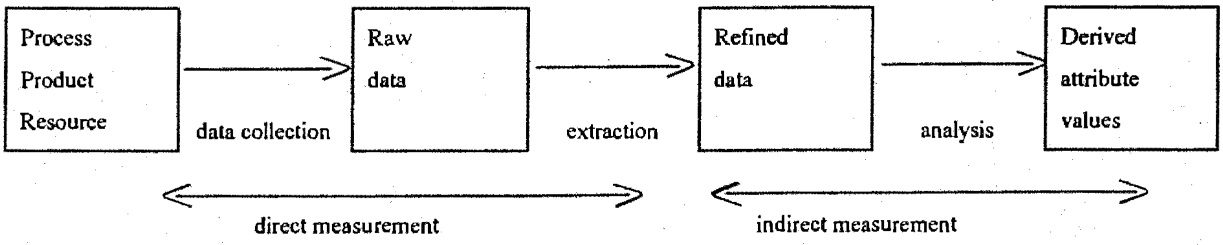


Figure 4.5 The role of data collection in measurement

Hetzel (1993: 39) states five measurement data collection principles:
It should be

1) *unobtrusive*

2) *automated whenever possible*

3) *based on clear and unambiguous, published definitions*

4) *validated as collected (as close to the source as possible)*

5) *saved as a repository and for future validation or analysis purposes.*

This is also echoed in the two principles given by Fenton (1991: 16). They are:

1) *It must be sufficiently simple so as not to disrupt the working patterns of anybody outside the software metrics team, and*

2) *The data must ultimately be included in a software metrics database.*

Collection and analysis of software metrics data, even from only one project, provides a company with new insight. It can serve as an initial baseline. As more data become available, these initial baselines can be checked and improved upon (Anderson 1990).

Companies usually do gather one or the other kind of raw data. In software organisations, it is likely that a software configuration management library and project cost information exist (Fenton 1991). This information can form the initial entries in a software metrics database. Daskalantonakis (1992) views a cost accounting system, a software configuration management system and a problem reporting/corrective action system that are in place in an organisation as prerequisites for collection of metrics data. He indicates that the existence of these systems will increase the likelihood of success of a software metrics program as it can facilitate the data collection and analysis process.

The resistance of managers and software engineers to collect data, mainly with regard to the time and labour needed to collect data, is a major stumbling block .

Pfleeger (1993) mentions two figures regarding the costs involved with data collection:

1) data collection and analysis add seven to eight percent to the cost of a project in the case of the Software Engineering Laboratory at the US National Aeronautics and Space Administration's Goddard Space Flight Center.

2) She quotes Tom DeMarco who, during the 1990 International Conference on Software

> Engineering, iterated his estimate that development costs increase between five and ten percent when metrics collection is involved.

Grady and Caswell (1987) also mention the following two elements that need to be taken into account regarding time and thus cost investment. Firstly, the time to

1) decide what data to collect,
2) decide which tools will help,
3) train the personnel involved.

Secondly, the ongoing investment in the actual data collection process. This is of importance to the project manager as it constitutes an overhead factor for the duration of the project.

A pre-operational investment in data-collection that will ensure compliance to the data collection strategy is thus essential.

Another aspect that is of primary importance is the accuracy of the data. As software development is a human intellectual activity, data collection requires human observation and reporting which is subject to bias, error, omission and delay (a time aspect which influences timeliness of data) (Kitchenham & Mellor 1991). Grady and Caswell (1987) note the example of programmers time where, in the case of HP, they had to trade-off accuracy versus the desire to get large numbers of projects to collect data.

The team responsible for implementing the software metrics program, is responsible for producing guidelines and counting rules for consistent data collection across the organisation (Grady & Caswell 1987; Daskalantonakis 1992; Pfleeger 1993). The public and private aspects of data will be discussed in 4.3.6.

The absence of historical data on projects has been cited as the main obstacle in software cost estimation. Software cost estimation and the role of historical data collection will be discussed in chapter 5.3.

Data collection thus is the backbone of any software metrics program. The planning and execution of this aspect will ultimately determine the success/failure of the program.

4.3.4.2 Why a company-wide database?

A prerequisite for the effective handling of the collected software metrics data is a company-wide database. Practical experience and insight regarding software metrics databases for three companies that have implemented software metric programs are described.

1 **Hewlett-Packard (HP) - Grady and Caswell (1987)**

The need for a company-wide database of the software metrics data to handle data from projects as well as process data was identified at HP.

Two approaches to a company-wide database were tried by HP:

1) the use of a network database manager on a multi-user system and
2) the use of a commercial spreadsheet program.

A disadvantage of the first approach was the inflexibility inherent in a network database. It was then decided to use a selected spreadsheet program (called the Software Metrics Database (SMDB)) as it has several advantages such as

1) it is easy to modify worksheets;
2) it has good graphics capabilities;
3) it is easy to use;
4) it is available to users in the company and
5) it requires minimal training.

Distribution issues such as anonymity (of project names and managers), security (to keep data internal to HP), the medium of transmission (via electronic mail network or floppy disks) and the aspects of updating data and identification of personnel to whom data will be sent have been addressed. These distribution issues are important to address in any company using a database to collect metrics data.

The following important uses of the database were identified:

1) it provides historical checks and can be used to double check an estimate at project level (by examining projects of similar type, size and language);
2) the known data from projects (elapsed engineering months invested and the elapsed calender months after each phase of development) is very useful to check (and help in determining) project estimates and it can be used at divisional level to compare progress.

Positive feedback was reported regarding the use of the database. It is used across the spectrum of metric users (software engineers, productivity managers and quality assurance managers). New areas of application are also experimented on by the users.

The use of a relational database with programmatic interface for tool integration was regarded (at the stage when the book was written) as the next step in the development of the company-wide software metrics database.

2 The Contel Technology Center - Pfleeger (1993)

Data collection has two legs in the software metrics project at CTC:
Project metrics databases and a Corporate historical database.

The aim was

1) to enable project managers to collect and use their own project data and
2) the gathering and analysis of the project data as part of a large corporate database.

The intent was to store the metrics both in individual project databases and in a corporate database. As the tools used by different projects could differ, the project manager was responsible for translating the data to a standard format (set out in a common counting scheme) for inclusion in the corporate database.

3 **Motorola - Daskalantonakis (1992)**

Daskalantonakis (1992) comments that there have been requests from within their company (Motorola) to centralize data in a company-wide database. Their approach has been that the metrics program is more manageable when it is initiated by encouraging decentralized data storage, analysis and feedback, so that the data is close to its source. He remarks that once the metrics program is well established, decentralized databases storing data from local projects can be connected to provide benchmarking data across the company.

Hetzel (1993) and Fenton (1991) also mention the establishment of a company-wide database as part of their data collection principles.

Additional benefits of a company-wide database (Fenton 1991) are:

1) provision of a common culture in the company
2) it raises the level of awareness of a metrics program and
3) improvement in the accuracy of assessments and predictions as the database grows.

A company-wide database is thus regarded as a very important aspect of Software Metrics Programs. The selection of an appropriate package to establish the database will now be discussed.

4.3.4.3 Package selection for the company-wide database

In the selection of an appropriate package for the database it is important to ensure compatibility and interface abilities with other software packages used in the company, as well as ease of automation.

Current commercial spreadsheet (LOTUS, QPRO, EXCEL) and database programs such as DBASE IV are mainly used by metric practitioners (Grady & Caswell 1987, Möller & Paulish 1993, Pfleeger & Fitzgerald 1991).

The Lotus 1-2-3 package was used for the project metrics databases at the Contel Technology

Center (CTC) (Pfleeger 1993). The information was used by project managers to make decisions regarding development and maintenance aspects. The package was chosen on the ground that it was already well-known by the managers involved. The software metrics toolkit (to be described in 4.3.5) transferred the inputs and outputs from the tools automatically to the spreadsheets. A postmortem analysis for finished projects was also envisaged. Additional important information from this analysis can then be added to the database. Pfleeger (1993) envisions a corporate database that would not merely be a collection of the individual project databases but a database-management system that can be used for more detailed analysis, e.g. the analyses of process information to evaluate general corporate trends.

Specialized packages have also been developed. As part of the ESPRIT[1] MERMAID project, a package called the M-BASE Data Collection and Storage System (DCSS) was developed at the National Computing Centre (NCC) in Manchester, United Kingdom.

The M-BASE package allows an organisation to *define its own data model(s) for data collection, that generated a database and input facilities to allow data to be collected against the defined models* (NCC Fact Sheet 1992: 9). An evaluation copy of this system was acquired through NCC (M-BASE ... 1993).

The system provides the following features:

1) *Support for defining a data model which identifies the attributes that will be collected on software development at project level, at pre-defined project milestones, and for project components (e.g. tasks, modules, staff, data, function, document).*

2) *Definition, typing and ordering of sequential milestones (e.g. problem definition, design, code, test).*

3) *Definition of component tables (permitting only key attributes per component).*

4) *Basic definitions for 131 attributes together with counting rules and definitions for 84 metrics.*

5) *The ability to define new attributes and metrics either from scratch or based on the pre-*

[1] European Strategic Programme for Research and Development in Information Technology

defined attribute and metric functions.

6) *Attribute viewpoints that give sub-setted lists of the attributes according to the type and feature being reviewed.*

7) *Suggested attributes are provided relevant to the different milestone types and component types. In addition, attribute views are available on attribute categories (e.g. size, cost parameters, personnel, product, process, change).*

In addition, the package provides:

1) *Automatic generation of a database to support the defined model.*
2) *Automatic generation of standard screen facilities with user-defined data validation.*
3) *Capability to interface to DBASE III compatible statistical packages.*
4) *File-based data entry from DOS text files in a defined column-based format.*
5) *Data and data model import, export and archive capabilities.*
6) *The generation of reduced functionality versions of the DCSS for data collection.*

Selection thus entails in-depth research into company strategies so that the package(s) used will optimise cost, efficiency and future return on investment.

4.3.5 MEASUREMENT TOOLS

Measurement tools play a core role in software metrics programs. Automation of software metric collection and analysis to ensure timely and cost-effective information, is one of the make or break aspects that determine the success of a software metrics program. Grady and Caswell (1987: 96) emphasize that *tools must be available to insure consistent measurements as well as to minimize interference with the existing processes of software development* in order to make software measurement successful. Tools for automating metrics are also considered as an important aspect of a software metrics infrastructure (Daskalantonakis 1992).

Effective tools are thus necessary to aid data collection in order to reduce time and costs involved. Software engineers need tools and techniques in order to minimize the time spent on collecting metrics (Pfleeger 1993). Grady and Caswell (1987) warn that time investment can be a problem, even with the availability of automated tools. They emphasize the need for a person or team to provide tools and training in the use and interpretation of the tools when and where it is needed in the organisation. If this aspect is not looked after, it will lead to inaccurate or, worse, no collection of data at all. To overcome this problem, the Hewlett-Packard organisation has created a job function of "productivity manager". The issue of tool availability is his/her responsibility. As these positions are in every division, it provided communication regarding best practises and tools across the organisation and helped to encourage usage of new technology faster throughout the company.

The metrics team, set up by the organisation to drive the software metrics program initiative, are the people responsible for researching, selecting and providing tools for personnel involved. Management may see the establishment of a group to select, develop and maintain the tools as an extra financial burden, but the economic benefit arising from this can be substantial (Möller & Paulish 1993).

Grady and Caswell (1987: 5) remark that one of the objectives of initiating the Software Metrics Council in HP was to establish a *measurement foundation against which the tools we planned to develop or purchase could be evaluated to determine their effectiveness.*

The Metrics Working Group in the Motorola company created the requirements for an automated metrics collection, analysis and feedback system and provided it to tool groups who were involved in automating software metrics. Criteria for evaluation of metrics tracking systems were also developed to facilitate the process of selecting commercially available metric tools. A list of existing metric tools were compiled by the group and send to interested metric users in Motorola (Daskalantonakis 1992).

The metric team at CTC had to provide tools to support metric collection and analysis at the project level (Pfleeger & Fitzgerald 1991; Pfleeger 1993). As their approach include the establishment of a metric tool kit, it will be discussed under the heading "A METRIC TOOL KIT" (4.3.5.1).

Pfleeger and Fitzgerald (1991) found that almost all the tools on the market could be categorised into two categories:

1) cost estimation and project management tools
2) code analysis and testing tools.

They identify the need for tools that address requirements-related metrics, process-related metrics and maintainability metrics.

Furthermore, Pfleeger and Fitzgerald (1991) state that tools differ widely regarding functionality, user interface and price. They identify the inability of tools to integrate with each other or with the software being developed as the main disadvantage of the metrics tools that are currently available. They also warn that the price of using metrics tools varies and is not directly proportional to the amount of functionality provided by them. If their costs are too high, the cost of the tool may outweigh the advantages of collecting and analysing the data within a project. They stress, however, that despite limitations, the tools provide valuable information about the software development process.

Möller and Paulish (1993) remark that many commercial tools exist, but are not widely used. They ascribe the situation to the fact that the tools address specialized metrics, and not the global indicators that are required by project management. The acquired tools need to support the software development process and have to be maintained and updated if the process changes.

The interdependence of certain metrics (e.g. time, effort and size) is an additional important consideration when evaluating metric tools (Möller & Paulish 1993) . Time, effort and size play a specific role as they are often used for normalizing purposes. Communication between chosen tools are therefore necessary. According to Möller and Paulish (1993), manual or poorly designed automatic transfer of files should be avoided. They give the following points of advise for the development of in-house tools:

1) use widespread commercially available packages (e.g. Lotus 1-2-3, Excel)
2) require the ability of the tools to process files produced by other tools, particularly standard ASCII files

3) require the ability of the tools to have output files that can be processed by other tools, (particularly output as standard ASCII files)
4) *give preference to spreadsheet programs for easy-to-use and isolated tools*
5) *prototype all the tools on spreadsheets for validation*
6) *give preference to the database with a programmable environment for an integrated metrics program at the business enterprise level.*

Möller and Paulish (1993) conclude that ease of data interface should be the main criterion when choosing metrics tools.

Grady and Caswell (1987) however, define metric tools in a wider context to include manual techniques such as paper forms. A metric tool that was developed, in Hewlett-Packard, for the presentation of the data in the form of useful graphs from a minimal standard set of data was called PM2L (Project Management Metrics Tool). It consists of an interface template to a commercially available spreadsheet. Data is entered and graphed weekly. A definite advantage of this tool was that the data could be manipulated and viewed in different ways. A second tool, to facilitate analysis of project completion metrics, was the SMDB (Software Metrics Database). This has been discussed in 4.3.4.2.

Grady and Caswell (1987) indicate that the objective of successful integration of software metrics collection and use in the software development process can only be met if tools for automatic collection of some metrics, simplified manual collection for others and flexible analysis of all data is available. They emphasize that total automation have some disadvantages, e.g. it can "freeze" useless measures into the process. Manual collection (e.g. forms, questionnaires) allows for experimentation, and although more labour intensive, may save costs in the long run as incorrect decisions regarding tools and measurements are less likely to occur. As accurate resource and process measures still depend largely on staff completing manual forms, Fenton (1991) emphasizes that forms should be easy to use and to validate.

The aspect of manual data collection is also discussed by Kitchenham and Mellor (1991). They suggest the following:

1) keep the procedures simple
2) avoid unnecessary recording
3) train staff in the need to record data and the procedures to be used
4) send feedback on data analysis promptly to the people who provided the data
5) validate the data.

Procedures for form design and handling also need to be established. It is necessary to define who fills in what, when, and where, and how the completed forms are to be processed. Interviews, as an additional means for data collection, have been mentioned by Conte (1986). It is however, costly, and can slow down the development process if not conducted carefully. The desirability of automatic data capturing to ensure accurate data collection is valid but the disadvantages discussed above need to be taken into account (Kitchenham & Mellor 1991).

Metric tools will ultimately be selected by the practitioner on the grounds of availability, cost, functionality and ease of interface abilities.

4.3.5.1 A Metric Tool Kit

A Metric Tool Kit was developed by the metric team of the Contel Technology Center (CTC) in response to requests from managers (Pfleeger & Fitzgerald 1991; Pfleeger 1993). The projects metrics database (discussed above) served as basis for the tool kit.

Pfleeger (1993: 70) explains: *The tool kit included metric tools to collect and analyze data appropriate for the project's process maturity and development environment and the project manager's needs and preferences. Based on an IBM PC, the metrics tool kit used several commercial tools and some in-house applications and spreadsheets. Underlying all applications was Lotus 1-2-3, which served as a unifier and acted as a repository for all measurements.*

The personal computer (PC) was chosen as platform for the tools because of its minimal cost and the great number of metric tools that are available to run on it.

The metric team at CTC added many Lotus 1-2-3 applications to the tool kit to simplify analysis.

The tool kit and its components comprised less than $2000 of commercial software and the team built and designed it in less than six months (Pfleeger 1993).

The tool kit was very successful. Metrics collection and analysis on every project was on the brink of becoming mandatory and the tool kit was used by four of Contel's major business units on their projects (Pfleeger 1993). Unfortunately the CTC closed due to GTE who bought Contel and disbanded the CTC.

As the existence of a metrics tool kit is seen as important with regards to the cost-effectiveness of a software metrics program, a summary of the steps needed to establish the tool kit are described (Pfleeger & Fitzgerald 1991).

The first step is the evaluation of the automated metric tools available, both commercial and those provided free to interested users. Tools are either "stand-alone" tools or embedded in CASE (computer-aided software-engineering) tools.

The first stage of this evaluation, called a paper evaluation, reviews the literature (including third-party evaluations in journals and trade publications) and documentation of the tools. It determines the intent of the tool, the type(s) of metrics it supported, its environment in which it is to be used, its interface abilities with other tools and the type of user interface provided by the tool.

Tools are then classified according to certain criteria. Pfleeger and Fitzgerald (1991) use faceted classification. Facets are defined as *multiple indices used to identify groups of similar objects. That is, each facet characterizes an attribute of the object that cannot be described using any of the other facets* (Pfleeger & Fitzgerald 1991: 479). Facets chosen by Pfleeger and Fitzgerald (1991) are:

1) **Type:** The type or purpose of the tool, e.g. a line counter.
2) **Activity:** Activity indicates the development phase to which the tool can be applied, e.g. design, code, testing etc.
3) **Level:** The minimum process maturity level at which the tool can be applied.
4) **Method:** The development method or model that the tool supports, e.g. the COCOMO

cost estimation model.

5) **Language:** The languages supported by the tool, e.g. the tool can analyze C and COBOL, but not ADA.

6) **Operating system:** The operating system that is required for the tool to run.

7) **Platform:** The hardware required for the tool to run.

8) **Target application:** The system type that the tool is designed for (e.g. management information systems).

This characterization makes it possible to describe every situation in which the tool can be applied. It thus allows for multiple descriptors for each facet. An additional benefit of this type of classification is that additional facets can very easily be added to the scheme. The only restriction that applies is that the new facet needs to be independent of any of the existing facets already included.

A database contains the tool evaluation information. Queries made by project managers in terms of the facets (by stating their requirements) enable them to read only those tool evaluations that apply to their situation.

The results of the first stage are then used to suggest a small subset of metrics tools that warrant further investigation based on the particular needs of the specific project.

The second stage involves the installation and use of the actual tool (a functioning version) with "real" data. Evaluation includes the examining of the speed (performance), data import and export capabilities, user-interface quality, documentation, vendor support, cost and tool accuracy (Pfleeger 1993).

The following metric tool evaluation form (figure 4.6) that can be used within a software organisation is adapted from Pfleeger and Fitzgerald (1991: 479).

Section 1 contains general information. Section 2 contains the classification of the tools according to the facets. The first part of section three (tool evaluation) contains the necessary information regarding version, platform and operating system. Subjective evaluation of the strengths and weaknesses is also included. The second part of section 3 contains a summary table of the objective evaluation of the tools.

1.0 THE TOOL

Tool name:

Vendor name:

Vendor address:

Contact person/phone/fax/e-mail:

Evaluation date:

2.0 TOOL CLASSIFICATION

Type	Activity	Level	Method	Language	Operating system	Platform	Target application

3.0 TOOL EVALUATION

Version:

Platform:

Operating system:

Cost:

Strengths:

Weaknesses:

CRITERIA	RAW SCORE (1-10)	WEIGHT (1-10)	TOTAL
Performance/ speed			
Data import/export			
User interface			
Documentation			
Tool accuracy			
Vendor support			
Cost			

Figure 4.6 Metric Tool Evaluation Form

An explanation of the above criteria follows:

1) Performance/speed: The execution time of the tool in performing calculations or analysis is rated.
2) Data import/export: It refers to the means used by the tool to import/export data from/to other tools (higher scores are obtained for simpler data transfer mechanisms).
3) User interface: The ease with which a user can learn to use the tool and the ease of use.
4) Documentation: The availability and quality of the documentation provided with the tool.
5) Tool accuracy: Rating given to judge the accuracy of the tool when implementing a model for a certain metric and its flexibility to provide modification of the parameters of the implemented model.
6) Vendor support: Rating on the vendor's provision of support (help lines etc.).
7) Cost: Criterion based on the cost to implement the tool on a company-wide scale.

Each possible rating (1 (low) to 10 (high)) in each category is described in detail in a set of tables, to ensure that no ambiguity between ratings exists. Pfleeger and Fitzgerald (1991) based the rating definition tables on tables reported in Bohner (1989) and Reifer (1986). Weights are assigned by the project manager doing the evaluation as different projects have different needs and different desirable characteristics. The final score is computed by multiplying the raw score by the assigned weight.

The information resulting from the evaluation form was stored in a database, providing managers with the necessary information to select tools. Process characteristics, such as the maturity levels are included in the database. The team uses the evaluation results to build a metric tool kit tailored to a project's need. An example is given in Pfleeger and Fitzgerald (1991). Thus, by specifying information about the project (environment, methods and metrics needs) the evaluation database can be used to suggest appropriate metric tools. The manager can thus base his final decision on development environment, the tool strengths and weaknesses, as well as the development process itself.

Furthermore, the existence of a tool kit cuts costs dramatically as no individual evaluations of tools need to be carried out. Suggestions to vendors regarding their product is another positive by-product of a metrics tool kit.

Updates and additions to the metrics tool kit was envisaged by Pfleeger and Fitzgerald (1991) as existing tools change and new ones are introduced. Users of tools will be interviewed and ratings will be changed to reflect experience with each tool. The section regarding strengths and weaknesses will also evolve as users become more experienced.

The main disadvantage of the tool kit approach was the time aspect involved in the customization of the tool-kit to the projects. Coordination between tool kits is another problem-area.

The author proposes that the evaluation form (figure 4.6) (Pfleeger & Fitzgerald 1991) can be used successfully within organisations as an instrument to evaluate individual metrics tools that they envisage to use. It is thus applicable outside the metric toolkit realm.

Stark and Durst (1994) also describe a metrics toolkit that were developed for the metrics initiative at NASA'S Operations Missions Directorate. Consistent data collection and ease of analysis were necessary prerequisites for assisting good decision-making. A standard set of tools were defined. The toolkit exists of a data repository element (a database or spreadsheet program), a cost/resource estimation tool, a size/complexity collection tool and a reliability estimation tool. The toolkit was not fully automated or integrated. However, all the components could share data. Costs were also low (less than $1000 and took less than a month to integrate and begin using). Project managers at MOD used it and found it useful. The toolkit also increased the availability of analysis of metrics options for project personnel.

4.3.6 "PEOPLEWARE" - THE HUMAN FACTOR

Metrics are never going to catch on as long as we are content to only use the metrics that are handed down from the top. We will only be successful when every person says 'I can't get this done... without using metrics' - and they start to believe that.

Words of an U.S. Air Force general (Clapp 1993).

The core role of the human in the success of any Software Metrics Program is echoed in the words quoted above. In implementing a software metrics program, one needs to be aware of potential human problems and how one can overcome them (Fenton 1991). Failing in this respect will lead to failure of the program.

The human issues involved are complex psychological issues. No attempt will be made to address these issues. However, observations from practioners regarding certain aspects of human involvement that are regarded as important will be provided.

There is a strong reluctance from the side of software engineers to be measured. Managers that embark on a metrics program need to work with the personnel involved to ensure cooperation and the commitment to collect meaningful data (Grady & Caswell 1987).

One of the most important aspects is the reassuring of the people involved that the data will not be used against them (Grady & Caswell 1987). Management displays the urge to use certain software metrics to evaluate people, i.e. as a performance instrument, rather than regarding it as a tool for process improvement. Furthermore, many studies have indicated the wide differences in programmer's productivity (Boehm 1981; Jones 1986). It is a potential dangerous step to apply metrics, especially in the early days of a metrics program, to assess performance. Grady and Caswell's experience at HP (Grady & Caswell 1987), leads to the following reasons for not using metrics in this way:

1) Measurement has not been done long enough to be certain of the accuracy of the measurements.
2) They have not established which metrics, or combination of metrics, correlate best with the behaviour they want to encourage.
3) The use of the metrics data as a performance instrument will lead to distortion of the data.

Additional factors, such as health, expertise, and the importance of a certain metric in measuring performance need to be taken into account if metrics are used in performance evaluation.

In training classes for metrics at HP, a major concern raised by both engineers and managers concerns the potential misuse of data. Grady and Caswell (1987) stress the fact that the approach of managers in interpreting metrics data needs to be nonthreatening. They emphasize: *software metrics today are not consistently enough defined and understood that anyone should consider using them to measure and evaluate people. Furthermore, premature usage of metrics data for such purposes will only cause future data to be distorted and useless* (Grady & Caswell 1987: 95).

Grady and Caswell (1987) give an example of legal action brought against a major company in Italy by the workers' union concerning the monitoring of data entry personnel. The extremes of human concern with regards to measurement were illustrated in this case.

Fenton (1991) also emphasizes that it is of utmost importance that the personnel need to be sure that the measurements will not be used to assess individuals. People fear possible "punishment" in the form of demotion or sacking as a result of the assessment. Accurate data that is not manipulated by individuals to their own benefit will only result if these fears are resolved. He advises monitoring the team instead of the individual but warns that there will still be a tendency to attempt to manipulate the results.

Möller and Paulish (1993), in addressing the issue regarding the use of metrics as a personnel performance measurement instrument, advise that the Personnel Appraisal System should be independent from the Software Metrics Program. They emphasize that the role of the metrics program is to help in improving the processes in the organisation. Assessing individual performance by means of the metrics will impact the program negatively. They stress that a well-established Metrics Program's information can help in assessing individuals but should only be used as supporting information.

Daskalantonakis (1992) also emphasizes the fact that concentration should be on process improvement instead of personal evaluation. Grady (1992: 120) advises that functional managers *need to build an atmosphere of mutual trust and respect for people's abilities to measure and understand the changes necessary to remain competitive.* Furthermore, he sees the project manager as a person who has the "best opportunity to understand the needs of both the organisation and the people".

Grady (1992: 120) suggests rules of etiquette (table 4.2) for applying software metrics by functional management, project management and the project team.

FUNCTIONAL MANAGEMENT	1. Don't allow anyone in your organisation to use metrics to measure individuals. 2. Set clear goals and get your staff to help define metrics for success. 3. Understand the data that your people take pride in reporting: don't ever use it against them; don't ever even hint that you might. 4. Don't emphasize one metric to the exclusion of others. 5. Support your people when their reports are backed by data useful to the organisation.
PROJECT MANAGEMENT	6. Don't try to measure individuals. 7. Gain agreement with your team on the metrics that you will track, and define them in a project plan. 8. Provide regular feedback to the team about the data they help to collect. 9. Know the strategic focus of your organisation and emphasize metrics that support the strategy in your reports.
PROJECT TEAM	10. Do your best to report accurate, timely data. 11. Help your managers to focus project data on improving your processes. 12. Don't use metrics data to brag about how good you are or you will encourage others to use other data to show the opposite

Table 4.2 Rules of etiquette for applying software metrics

Additional workload as a result of the software metrics program, is another human obstacle (Fenton 1991). People would not like to participate if they have to do additional measurement work on top of their busy schedules. It is very important, from top management side, to acknowledge the fact that extra resources are required for the successful implementation of a software metrics program. The adverse effect, in terms of resentment, that can be caused by people given responsibilities that have previously been the responsibilities of staff now busy with the metrics program needs to be avoided. Ideally, new staff should be appointed.

The "Hawthorne Effect" (named after the Western Electric plant in Hawthorne, New Jersey, USA, where experimentation in the 1920's first revealed this phenomenon), must also be accounted for (Fenton 1991). It implies that the very act of measuring leads to the improvement, because people know that they are being observed. The question raised in Conte (1986: 24): *Is it possible to observe, measure, and quantify any activity without altering it somewhat in the process?* remains valid and must be closely monitored.

Anonymity of data is another aspect involving humans. Total anonymity is often impractical. Fenton (1991) advises the following regarding anonymity:

1) retain individual anonymity
2) retain complete anonymity if the metrics are only used for assessment
3) impose anonymity if the data is being inspected or used by departments who are not involved in the original objectives of the program
4) give participants in the program the option of not remaining anonymous if anonymity is possible.

Grady (1992) discusses this aspect in the context of private versus public data. He gives the example of defects in the software. Personnel developing software like to keep defects private. However, after delivery of the software, bugs are found. It then becomes public. Grady (1992:104) points out that they try to instill *an attitude towards problem-solving rather than finger-pointing* in people. He mentions the importance of inspection, where defects are found by the inspection team, making it public to them, but not to the customer or even to other project teams in the organisation. The "blame" also shifts from the individual to that of the team responsible for developing that particular module of the software. Teams, however, are also prone to sensitivity, especially with regard to time data. Grady (1992: 105) also mentions the aspect of information hiding, which, in the context of developing software, means *a software module should only provide information at its interfaces that other modules require to do their job correctly.* Information hiding is negative in the instance where it is a result of inadequate planning, or is a way to conceal relevant management issues. He concludes that data that is typically private to a project team includes *detailed estimates and actuals of number of modules, size and complexity of modules, and projections for how many defects will be found and when* (Grady 1992: 106). The project team thus feel that they have ownership regarding the use and interpretation of this data. Grady (1992) mentions that the metrics that are public to the organisation are calender times, defect rates, project costs, and some measure of functionality of the products. Applying the principle of private/public data, will help the personnel involved in an organisation to determine *who should have access to what data and how knowledge of the data should be applied* (Grady 1992: 107).

As any metrics program will introduce change in the organisation, Grady and Caswell (1987: 92) apply four aspects that threaten individual identity in a change process, to their metrics program. They suggest that people can be expected to react as follows:

1 *MEANING (What is the personal significance of a change?)*
People will resist the extra duty and time to collect data. They would want to know how their performance is going to be measured by the data.

2 *MASTERY (How can an individual regain control of a situation?)*
Collection of metrics may be resented if the individual feels that he/she has no control over it. The person concerned will strive to use the data to reflect effort positively and to prove his/her own points of concern regarding an aspect in the work environment.

3 *MERIT (What is a person worth under the new circumstances?)*
People will support metrics that they feel will emphasize areas of performance that they are proud of and will attempt to make these ones more important.

4 *MORALE (What difference does it make whether a person tries or not?)*
The ground rules for interpretation of data need to be known beforehand.
It would lower the morale if a person thinks it is going to measure how poorly he/she perform.

Any organisation that implement a software metrics program needs to be aware of these factors and address them beforehand.

Interesting factors that have been observed by Grady and Caswell (1987) where metrics were used successfully by a team were the following:

1) The means of measurement were easy and were well understood. The effort involved in measuring was minimal.

2) The team was measured, not the individual.

3) The team agreed beforehand that the measurements were meaningful.

4) Data was showed publicly, as the team go through the process before committing to a schedule.

Eventual involvement of all people using meaningful measures is necessary for the success of a software metrics program. According to Grady and Caswell (1987), the people closest to the process are the ones who can most quickly help to bring it under control, and measurements will help them to identify how. Pfleeger (1993) emphasizes that the people involved would only collect and analyze metrics correctly when the metrics meet a specific need or answer an important question. The most common cause of complaint regarding metrics arises when metrics that were gathered for a specific agreed objective in mind, are used for a different non-agreed objective (Fenton 1991).

Daskalantonakis (1992) identifies different groups of metric users and their principal interest (in brackets):

1) software users (quality and value of software product)

2) senior managers (overall control and improvement across projects in the business unit/company)

3) software managers (control and improvement of projects that they are responsible for)

4) software engineers (control and improvement of specific software project activities and work products in which they are involved)

5) software process engineers and software quality assurance team (cross section of the previous four users, depending if they work at the business unit/company level or at project level).

He also acknowledges the needs of these different types of users and gives the following aspects as requirements for addressing the needs of the metric users:

1) define metrics and obtain consensus/acceptance by the users involved (discussed in 4.3.3)

2) train metrics users and provide consultation support (to be discussed in 4.3.7)

3) automate the data collection, analysis and feedback process (as discussed in 4.3.4 and

4.3.5).

It is clear that no software metrics program will succeed without the cooperation of the personnel involved. It is people who collect, interpret and "own" metrics data. Sensitivity at all organisational levels should exist and must be recognised and accommodated (Grady 1992). It is critical to take note of the human issues before the implementation of the program (Daskalantonakis 1992) and to be sensitive regarding issues that the personnel involved raise regarding the metrics program once it is started.

4.3.7 TRAINING AND CONSULTING SUPPORT

The issue of training runs like a golden thread through the previous implementation aspects that were discussed. It is the thread that enables all the other aspects to function and to form a successful whole.

The success of a software metrics program is dependent upon the support by the workforce, as discussed above. Support can only be attained through training and thus motivating people to cooperate.

Training should be preceded by presentations (as mentioned in 4.3.3.1), to "sell" the idea of a software metrics program and the benefits of such a program, i.e. the why, what and who aspects of software metrics (Grady & Caswell 1987). This increases the awareness of the need for software metrics.

Grady and Caswell (1987) identify the need for training to provide engineers and project managers with detailed knowledge and skills for effective and accurate data collection. Specific training is required when an identified set of measurements is to be used within a project(s). The aspect on "how" to measure and the tools that are to be used have to be explained (Fenton 1991).

The course objective for training in software metrics in the Hewlett-Packard company (HP), according to Grady and Caswell (1987: 175), was: *To provide background and hands-on experience to project managers and engineers so that they can immediately use software metrics* *in their own environment to make informed decisions in the software development process.* With this aim, effective learning could take place. An outline for the HP software metrics course is provided in Grady and Caswell (1987: 76).

An important success factor of the training courses in the HP case was that course outlines and implementation plans were discussed with representatives beforehand, making it acceptable and directly applicable. Another success factor, according to Grady and Caswell (1987), was follow-ups and consultation support by the initial trainer and course developer for his past students.

Daskalantonakis (1992) views training and consulting support as an integral part of the software metrics program. Through the Metric Working group in the Motorola company, a two-day training workshop has been developed and has been taught across the company. He also reported success as a result of hands-on consulting activities by the instructor to participants, noting that it provided an effective mechanism for software technology transfer.

Daskalantonakis (1992) recommends the use of an external consultant early in the project to initiate data analysis for process improvement and process control. However, after this, the engineers and managers involved in the project, are to analyse and interpret the data as they have expertise and knowledge pertaining to the project.

Fenton (1991) recommends training classes, that address a range of software measurement aspects, as a regular staff development feature.

Graphs are an excellent medium to be used in training and are easily understood. The use of graphs in software metrics presentations and training is prominent in Grady and Caswell (1987); Grady (1992) and Daskalantonakis (1992).

Training is thus a necessary part of the software metrics program and its value should not be underestimated. Consulting support, as a complement to the training program, will ensure ongoing support for the metrics program as people will be kept well-informed and up to date. Lack of support for the software metric program due to problems encountered, can be alleviated by means of training and consultation.

4.3.8 IMPLEMENTATION PROBLEMS

As with any new initiative in an organisation, implementation problems are encountered.

Möller and Paulish (1993) list the following implementation problems regarding software metric programs (each problem will be briefly described):

1 **Lack of Acceptance**

Reasons that are often given for the lack of acceptance are: metrics may restrict the process of creativity; metrics will lead to an additional workload; the benefits of using metrics are not clear; the human fear of being measured; and the difficulty in admitting that process improvement is necessary.

Möller and Paulish (1993) comment that this problem can be overcome by "selling" (explaining the goals and benefits) the concept of a software metrics program successfully through presentations and training to the entire organisation.

2 **Personnel Appraisal**

This aspect concerns the fear of people that the metrics will be used to measure their own performance and not organisational performance. This aspect has been addressed in 4.3.6.

3 **Quick Fixes - Unrealistic Expectations**

As emphasized in the introduction to this chapter, a metrics program can not be used as a quick remedy to large quality or productivity problems. Möller and Paulish (1993) indicate that an average of two years is required to notice benefits arising from the program. They add that companies where these practises have been quoted as "best practises", have had metrics collection and process improvement implemented for ten or more years. It has become part of the corporate culture and procedures. A multi-year period must be agreed upon by management for continuing the metrics program.

4 **Loss of Momentum**

It is often the case that after initial implementation, enthusiasm and motivation fade. Patience and good leadership is essential to maintain momentum. By focusing on weak spots, indicated by metrics, successful organisations have incrementally improved their processes and maintained momentum.

5 **Tools Availability**

Resources are required to select, develop and maintain tools as well as to provide training to staff in the use of tools. Management often see this as an additional financial burden, but the economic benefit of a good team to handle this aspect can be substantial.

6 **Management Support**

Visible support from management is essential for the success of a metrics program. Actions by management must illustrate their dedication to the metrics program.

7 **Poor goals or Follow-Up**

Metrics programs can fail if goals are not well defined or monitored regularly. Further, the implementation of actions as a result of indications by metrics need to be planned, organised and monitored. They emphasize that resources need to be planned and allocated for personnel, tools and equipment to accomplish the actions.

8 **Lack of Team Players**

Cooperation is required for measuring and improving the development process. Möller and Paulish (1993) emphasize the role of shared values and attitudes that is necessary to build a positive quality culture. They share the view that management should be a role model in this aspect.

The above-mentioned problems are echoed in Verdugo's [reported in Jeffery & Berry 1993] list of reasons for software metrics program failures:

1) *Lack of clear definition of the purpose of the program.*

2) *Personnel resistance due to perception of it being a negative commentary on their performance.*

3) *Data collection burden was added to already burdened staff.*

4) *Program reports failed to generate management action.*

5) *Management supports withdrawn because program seemed problematic and generating "no-win" situation.*

Another problem is misleading data that is collected because of inconsistent definitions. Out-of-date metrics has been a problem on other projects (Clapp 1993).

Awareness of potential implementation problems thus enables one to avoid possible software metrics program failure.

4.3.9 EVALUATION AND FEEDBACK

Evaluation and feedback mechanisms need to be established in order to enable the modification of the software metrics program. Mechanisms currently used are: seminars; workshops; publication of results in-house; training and meetings of the metrics team and the personnel involved in the data collection process. These activities will lead to the natural evolution of a software metrics program over time.

Jeffery and Berry (1993: 29-30) suggest assessment criteria for the four perspectives, discussed in 4.3.1, to evaluate and predict the success of a measurement program in an organisation. The assessment criteria are covered by the following questions:

1 CONTEXT

C1. Were the goals of the measurement program congruent with the goals of the business?

C2. Could the measured staff participate in the development of the measures?

C3. Had a quality environment been established?

C4. Were the processes all stable?

C5. Could the required granularity be determined and was the data available?

C6. Was the measurement program tailored to the needs of the organisation?

C7. Was senior management commitment available?

C8. Were the objectives and goals clearly stated?

C9. Were there realistic assessments of pay-back period?

2 INPUTS

I1. Was the program resourced properly?

I2. Were resources allocated to training?

I3. Were at least three people assigned to the measurement program?

I4. Was research done?

3 PROCESS

A PROCESS MOTIVATION AND OBJECTIVES

PM1. Was the program promoted through the publication of success stories and encouraging exchange of ideas?

PM2. Was a firm implementation plan published?

PM3. Was the program used to assess individuals? (Demotivating)

B PROCESS RESPONSIBILITY AND METRICS TEAM

PR1. Was the metrics team independent of the software developers?

PR2. Were clear responsibilities assigned?

PR3. Was the initial collection of metrics sold to the data collectors?

C PROCESS DATA COLLECTION

PC1. Were the important initial metrics defined?

PC2. Were tools for automatic data collection and analysis developed?

PC3. Was a metrics database created?

PC4. Was there a mechanism for changing the measurement system in an orderly way?

PC5. Was measurement integrated into the process?

PC6. Were capabilities provided for users to explain events and phenomena associated with the project?

PC7. Was the data cleaned and used promptly?

PC8. Did the objectives determine the measures?

D PROCESS TRAINING AND AWARENESS

PT1. Was adequate training in software metrics carried out?

PT2. Did every one know what was being measured and why?

4 PRODUCTS

P1. Were the measures clear and of obvious applicability?

P2. Did the end result provide clear benefits to the management process at the chosen management audience levels?

P3. Was feedback on results provided to those being measured?

P4. Was the Measurement system flexible enough to allow for the addition of new techniques?

P5. Were measures used only for pre-defined objectives?

A criteria scoring scheme was developed by Jeffery and Berry (1993) to measure success. Equal weighting was applied to the criteria. The criteria scoring scheme is:

0 - did not meet any of the requirements

1 - met some of the requirements

2 - met most of the requirements

3 - fully met the requirement

They applied their assessment criteria to three organisations. After their study of the three organisations, additional criteria were proposed that can be added. They are:

Context:

1) Identify who has the responsibility for obtaining benefits from the measurement program.

2) Ensure that management experience and training are sufficient to use the measured products.

3) Build a participatory management style.

4) Ensure a supportive industrial climate applies.

5) Ensure the level of technical difficulty ... is within the capacity of the software developers.

Inputs:

6) Use external consultants where needed to get additional experience and authority.

Process:

7) State the criteria at the onset of the program for evaluating program achievements.

Products:

8) Ensure chosen metrics are relevant and acceptable to target community.

The second additional proposed criterium, 2) above, is also emphasized by Clapp (1993). She points out that it has taken a long time for managers in government and industry to recognize the value of metrics data and to take the time and effort to both generate and analyses it.

4.3.9.1 Lessons learned: An overview of factors listed by practitioners

Lessons learned by organisations that have implemented software metrics programs are important feedback instruments. They provide organisations that embark on a software metrics program with valuable pre-implementation knowledge.

Grady and Caswell (1987) cite, based on their experience at HP, clear communication of metrics successes and overcoming the fear of measurement as the two main aspects that need to be achieved to ensure widespread acceptance and use of metrics.

Pfleeger (1993) lists the following themes that contribute to the success of the software metrics program at the Contel Technology Center (CTC):

1) *Begin with the process.* Derive the metrics from the process and its inherent problems. Developers are more enthusiastic when they see the connection between their problems and the data they are collecting.
2) *Keep the metrics close to the developers.* The project personnel themselves should be able to access and evaluate the metrics and take action as a result. This will enable them to make metrics-based decisions about the product or process effectively.
3) *Start with people who need help, then let them do your advertising for you.* By using

projects that had problems as a beginning, the collection of metrics was seen as welcome assistance and not as an additional burden. Success stories spread and make other project managers eager to participate.

4) *Automate as much as possible.* Minimize time spent on collecting and analysing metrics by using appropriate and cost-effective tools.

5) *Keep things simple and easy to understand.* Developers only need to know the relationship between the measurements they are collecting and the problems to be solved.

6) *Capture whatever you can without burdening developers.* The advise is to capture as much as possible, quickly and as unobtrusively as possible.

7) *If the developers don't want to, don't make them.* If developers do not want to collect a certain measure, do not force it. It will result in inaccurate data.

8) *Using some metric is better than using no metrics.* The biggest problem in establishing a software metrics program is to convince developers that the collection of metrics is worthwhile. Pfleeger (1993) advises that it is better to start with a small set of metrics.

9) *Use different strokes for different folks.* The metrics collected should reflect the project's process maturity and needs. Projects problems should be solved first, with organisational problems later.

10) *Criticize the process and the product, not the people.* People distrust metrics if they think they are going to be used as a performance measurement instrument.

Daskalantonakis (1992) echoes some of the themes mentioned by Pfleeger (1993) when he shares some of their implementation experiences at Motorola, as listed below:

1) It is better to start with a small set of metrics that address important improvement areas and evolve the metrics over time.

2) As managers and engineers begin to see the benefits of metrics, they explore new ways to obtain even more benefits.

3) A recent survey in Motorola indicates that a package that defines metrics and processes for the formal software review and testing process is used by a high percentage (67%) of software engineers and managers that were surveyed. The package is tailored for the different user groups and training material has also been developed and used.

4) As mentioned in the discussion on a company wide database (4.3.4), Motorola's approach

was to encourages localized data storage and analysis, keeping the data close to where it comes from until the metrics program is well established.

5) Project team members should be able to continue data collection, analysis and feedback once the metric team, and possible an external consultant, have set up these activities.

6) Motorola have requests from projects to collect only one metric in order to keep costs down. This is however, detrimental, as one can manage to optimise the result and other, more pressing problems, are not addressed.

7) The cost aspect of a software metrics program. Motorola's benefits through quality, productivity and cycle-time improvement were found to be worth the investment made.

8) The data has helped the project team to understand the extent of their problems. It motivated them to improve.

9) The metrics have helped to establish baselines, and to focus on actions with quantifiable results.

10) The quality initiative taken as a result of the analysed data made the difference, an aspect also emphasized by Möller and Paulish (1993).

Grady (1992) cites the following aspects, five-years after the experience at HP with their software metrics program was recorded in Grady and Caswell (1987):

1) A metrics program needs to start with a basic set of "primitive" metrics.
This helps in establishing a foundation from where one can later move to more complex types of metrics.

2) The importance of "selling" the concept of metrics. The strategy used was to focus on the use of metrics to track progress and identify improvement, not as an instrument to predict.

3) Some groups within HP tried to change too quickly. This results in collecting data without clear goals and objectives.

4) Too much attention to just one metric leads to biased data and thus, poor decisions.

5) The pressure for breakthroughs. Metrics data is valuable for problem detection, but the actions to resolve the problems are usually not simple or inexpensive.

6) Changing business conditions can pose a threat to a metrics program.
Metrics need to be integrated enough into an organisation to ensure that the program will not be discontinued due to new priorities.

7) Good tool support is a success factor in a metrics program.

Lytz (1995) shares the experience at Boeing, where they have introduced a fairly elementary software metrics program with the development of the Boeing 777 within the Boeing Commercial Airplane Group (BCAG). This group is the operating branch of the Boeing Company which designs, produces and markets all commercial Boeing jet transports.

1) The discussions that have been a consequence of the metric data have been more important than the data itself.
2) An effective software metric program would probably not have started without the pressure from top management.
3) *Involvement of the material organization (the business interface with the suppliers) was essential to make the metric programme work.*
4) The metric programme was started after the award of supplier contracts. It proved to be easier than expected, but there is agreement that it would have been better to start metrics prior to the award of the contracts.
5) Simple definitions used for code size and design completions were adequate.
6) The use of a simple, spreadsheet-based tool for metric tracking has worked well.

Clapp (1993: 108) provides six principles that "Software Management Metrics" (Schultz 1988) are based on:

a) A successful software development project is one that meets its cost, schedule and quality goals.

b) Development plans should set quantitative goals so that you can tell if you are meeting them.

c) Plans should be compared with actual performance throughout development to detect potential problems early.

d) Data trends over time are often better indicators of potential problems than the actual values, because they can show when deviations from the plans are temporary, fluctuating, growing or diminishing.

e) There are many explanations, good and bad, for the same set of data; metrics indicate not problems, but data values that should be investigated to see if there are problems.

f) The presentation of metrics can obscure or clarify their message.

It is important for the metric team members at an organisation to take note of these lessons. By knowing beforehand what type of problems have been experienced, planning could be directed to avoid, or at least, address those problems. Positive success aspects, on the other hand, can be followed and emphasized.

4.3.9.2 Evolution of a metrics program

Through feedback and evaluation by metrics users, the set of metrics as well as the program's infrastructure will evolve over time.

Grady and Caswell (1987) describe the requirements that need to be addressed once a metrics program is established in order to have a mechanism for maintaining a standard and to communicate successes and failures. In their case, the HP Software Metrics Council were responsible for:

1) Changes to and approval for software metric standards.
2) Research and publication of information and results within the company.
3) Enthusiasm for metrics and selling of metrics concepts.
4) Active involvement in software process improvements.

Grady and Caswell (1987: 184-185) also mention that the software metrics program continuation relies on "written feedback, personal contacts, group presentations at all levels, tool development, training, and by providing forums for sharing success stories".

Cox [reported in Fenton (1991)] describes Hewlett-Packards' new measurement activities that evolved as a consequence of deficiencies in the original database. The approach that has evolved is to differentiate between three levels of measurement: high level measurement (for group managers that needs strategic measures), middle level measurement (for division managers) and low level measurement (for project managers).

In his discussion on the evolution of the HP's software metrics program, Grady (1992: 206) provides a hierarchy of metrics acceptance and practise that they have observed at HP:

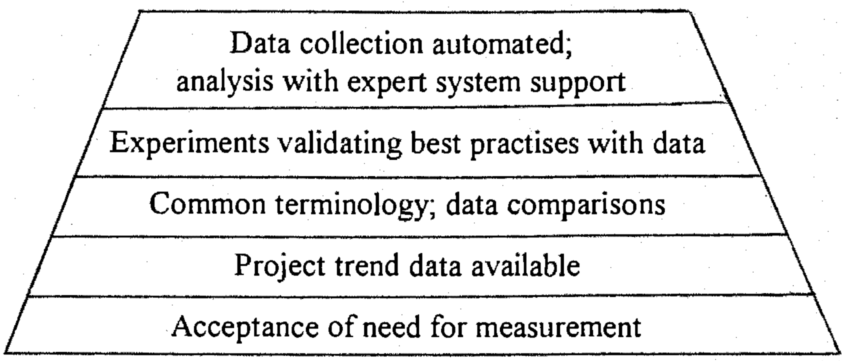

Data collection automated;
analysis with expert system support
Experiments validating best practises with data
Common terminology; data comparisons
Project trend data available
Acceptance of need for measurement

Figure 4.7 A hierarchy of metrics acceptance and practise

An organisational infrastructure that encourages metrics usage and sharing of results have evolved in the HP company's case (Grady 1992).

Feedback and evaluation at regular intervals will thus signal problem areas of implementation and ensure continuity and maturing of the program.

4.3.10 CONCLUSION

The implementation of a software metrics program is a complex undertaking. A practical approach to measurement, taking into account the topics that were discussed, is essential to enable successful implementation of a software metrics program.

Three factors identified by Ruben [reported in Fenton (1991)] as defining "success" of software are:

1) The results from the metrics program (refined data) are actively used in decision making.
2) The metrics program lasts longer than two years.
3) The metrics program results are communicated and accepted throughout the company.

These factors are the ultimate test as to the successful implementation of metrics.

4.4 THE STATE OF METRIC PROGRAM PRACTISES GLOBALLY

Hetzel (1993) discusses the state of metric practises worldwide. He mentions the Baseline Measurement Practises Survey, conducted in 1990 by Software Quality Engineering. It was a large-scale survey, distributed to eight hundred software organisations around the world. Its aim was to measure how industry was using software measurements and to benchmark what the best companies and projects are doing. It was found that company practises were highly variable. Overall usage of a representative list of selected measures was low. Another disturbing factor was that most organisations reported general dissatisfaction with their current measurement program. The baseline survey also confirmed that most measurement programs are in their early stages.

Hetzel (1993) also presents results from a survey on the use of 65 commonly cited measurements by attendees at the 1991 Applications of Software Measurement Conference. The purpose of the survey, according to Hetzel (1993: 8), was to *determine software measurement usage and perceptions of value from experienced and knowledgeable organisations and individuals.* Results regarding the program's maturity and effectiveness were harsh. Measurement program effectiveness was rated "poor" by 66% of the respondents and maturity of the program was rated as "in infancy" by 64% of the respondents.

Fenton (1991) gives a number of reasons for the relatively slow growth of software metrics programs in industry:

1) Disagreement between workers in the field (researchers and practioners) about the value of some of the proposed measures.
2) The cost to implement a software metrics program.
3) The extent of automation for the collection of metrics. Tools are required to address real industrial needs.
4) As a result of an application and maintenance backlog, developers can still make huge profits without the use of new technologies.

5) Material aimed at the practioners regarding software metrics are lacking.

6) Industrial software quality systems are still primitive.

It can thus be seen that software metric programs are still in their infancy and that a long road with many challenges is lying ahead. The survival of these programs will be greatly dependent on how the software community perceive them. Unless measurement is seen as an important element in any decision and vital and useful for everyone in the software community (Hetzel 1993), the success of a software metrics program is questionable.

4.5 EXTENDING THE METRIC APPROACH TO OTHER INDUSTRIES

The metric approach can be extended to that of other industries. Grady and Caswell (1987) give the example where a metrics program was planned for other engineering development processes in Hewlett-Packard. The metrics that they have chosen were divided into three categories: project/product metrics; process metrics and people metrics.

Stout (1992) mentions the use of metrics in the telecommunications industry by the Alcatel Network Systems (ANS) company in America.

In order to help manufacturing management to improve manufacturing performance, a proactive approach is required to ensure that manufacturing has the tools needed to make decisions that lead to continuous improvement. Schmitthenner (1993) suggested the use of metrics to help in improving manufacturing performance. He argues that financial statements are of little use in helping manufacturing management and that accountants' idea of important factors differ from that of manufacturing people. He gives an example of metrics (in the form of graphics) that are used at the Soladyne Division of Rogers Corporation in the United States of America. They have three categories of metrics, namely Customer Satisfaction metrics, Manufacturing Volume metrics and Manufacturing Performance metrics. He emphasizes the following aspects regarding the usefulness of the metrics:

1) The development of metrics is an ongoing process.

2) Use the "right" language (the metrics need to be understood by the people using them).

3) Focus on the metrics that improve profits.

4) The metrics defined need to be controllable by the department using them (e.g. manufacturing need to be able to control the metrics designed to improve manufacturing performance).

5) Make the metrics visible and in graphic format.

6) Make the information timely.

7) Use available data. Schmitthenner (1993) indicates that companies usually do not have to set up a new data collection system to collect the data required to provide useful metrics. Raw data is usually available but difficult to get at.

8) Determine the needs from the people that are going to use the metrics beforehand.

9) Take a macro (overall) view of the business.

Schmitthenner (1993: 30) concludes: *The thirty minutes or less taken each week to produce the graphs will do more to help the manufacturing teams than a year's worth of financial statements.*

Key performance indicators, generally known as KPI's, are metrics. They are indicators that are used to monitor and record the cost effective application of resources and the economic and physical performance of complex industrial operations. It can provide a company with "visibility" throughout. All levels of operational, administrative and support services are included.

Commonly, key business factors in Financial, Engineering, Logistics, Human Resources and more recently the aspects of social responsibility and environmental issues, are monitored and performance is reported against defined internal, comparative or international standards. Data may be recorded as raw or "normalised" weighted indices or ratios of data existing in one or more disciplines or operating sectors. There are usually four levels of information requirements:

Policy (5 to 10 year planning horizon and controlled by the year);
Strategic (1 year planning horizon and controlled by the month);
Tactical (1 month planning horizon and controlled by the week) and
Routine (1 week planning and controlled by the day).

A drawback of KPI monitoring and reporting systems was the amount of manual labour to complete and present the information. Fortunately, modern information systems can now provide the data to support activity-based KPI systems.

The MINCOM company has developed KPI templates for major industry segments. They are available on the spreadsheet program, EXCEL, and are planned for release on EIS (Executive Information Systems) such as FOCUS. These templates provide a visual framework for the KPI's and easy access to graphical representation of performance achievements (MIMS KEY PERFORMANCE INDICATORS 1993).

They stress that the most meaningful Key Performance monitoring results occur when:

1) the chosen performance indicators are vital to a corporation's success
2) the upper and lower performance measurements are accurate and
3) if used, the weighting (in relation to an indicator's contribution to the performance in question) that is applied is appropriate.

4.6 SUMMARY

Software metrics programs is an application of a holistic measurement approach to quality. It represents a long-term management commitment to understand and manage software development better. It is a clear example of management by fact.

The chapter has

1) familiarized the general reader with the software metric concept and software metrics programs in order to stimulate the possible use of such programs in other industries. The extension of the metrics approach to other industries and its equivalence to key performance indicators is briefly discussed.
2) cleared the definitional aspect of software metrics and related terms

3) identified and discussed two different software metric program approaches: the global and the project-oriented approach

4) proposed a practical framework to plan and develop the process of metric collection that can be used with each of the above-mentioned approaches

5) emphasized the critical role of accurate, on-time and sufficient data collection and the need for a company-wide database

6) suggested that the evaluation format in figure 4.6 can be used as an instrument to evaluate metric tools that an organisation wants to use.

5 STRATEGIC MEASUREMENT ISSUES IN SOFTWARE

"You ought to have finished," said the King. "When did you begin?"
The Hatter looked at the March Hare, who had followed him into the court, arm-in-arm with the Dormouse. "Fourteenth of March, I think it was," he said.
"Fifteenth," said the March Hare.
"Sixteenth," said the Dormouse.
"Write that down," the King said to the jury; and the jury eagerly wrote down all three dates on their slates, and then added them up, and reduced the answer to shillings and pence.

Alice in Wonderland (Lewis Carroll 1865)

5.1 INTRODUCTION

Two aspects that are crucial to improved quality and productivity in software are software reliability and software cost estimation. The latter does not come as easy as for the jury in *Alice in Wonderland.* Brettschneider (in Sheldon et al. 1992) states that, in addition to the prime concern by customers that software is too expensive, another major concern is that software is frequently unreliable.

The level of quality required, the time of delivery and the cost are thus the most significant requirements of the software user from the software producer. Quality, time and cost constitute the three dimensions of software development. Musa, Iannino and Okumoto (1990) remark that software quality, in the absence of a "concrete" measure thereof, has suffered against cost and schedule. As software reliability is a critical dimension of software quality, and quantifiable, it is of core importance. Sheldon et al. (1992: 13) remark that software reliability measurement *has become a significant factor in quantitatively characterizing quality and determining when to release software on the basis of predetermined reliability objectives.* The interaction between the three dimensions, e.g. poor reliability means additional testing and therefore cost, is extremely important and needs more investigation. It can possibly be investigated by means of a formal decision-making approach but will not be addressed in this document.

Increased complexity, in synergy with development and cost constraints, demand the need for measurement and prediction of software process and product characteristics (Musa et al 1990). This is echoed in the words of Basili and Musa (1991: 9): *In the 1990's, market forces will drive software development into quantitative methods for defining process and product quality.*

Measurement has been discussed from an organisational viewpoint in chapter 4.

This chapter will briefly introduce software reliability measurement and modelling to provide the reader with an overview of what it entails. Software cost estimation will then be described and discussed. In particular, two aspects of current interest will be addressed, i.e. the nonlinearity/linearity of software cost estimation models and the link and relationship between software cost estimation models and project management techniques such as PERT.

5.2 SOFTWARE RELIABILITY

5.2.1 INTRODUCTION

The impact of software failure as a result of poor reliability is severe and can often be critical. In medical and military systems it can mean the loss of human life. In a business or governmental situation, it can ultimately influence the difference between staying in power or not.

Software reliability is important in every stage of software development, that is, in requirements; design; coding and planning for testing.

Software reliability is described as a "measure" of how well the software functions to meet the requirements of the customer (Musa et al. 1990). They suggest that reliability is a much richer measure, than say, defect density, as it encompasses the user as well. It is not only a development-oriented measure. Pfleeger (1992) also emphasizes that two differing viewpoints regarding software reliability exist, namely reliability from the perspective of the software developer and reliability from the perspective of the user (customer).

5.2.2 DEFINITIONS

The following definitions are important within the context of software reliability.

5.2.2.1 Failures, faults, errors and defects

A **software failure** is defined by Musa et al. (1990: 8) as the ***departure of the external results of program operation from requirements.***

It is a dynamic definition and it is not a fault, or "bug" in the program.

A **fault** is defined by Musa et al. (1990: 8) as ***the defect in the program that, when executed under particular conditions, causes a failure.***

A fault is thus the commonly referred to "bug", an error of the programmer.

In addition, the IEEE/American National Standards Institute (ANSI) Standard 982.2 makes a distinction between errors, faults, defects and failures (Pfleeger 1992: 57). The definitions are as follows:

1 *Error*

Any human mistake that results in incorrect software; errors include an omission of a critical requirement in a software specification, a developer's misinterpretation of the requirement, or an incorrect translation from design to code.

2 *Fault*

An error's manifestation in software that causes a functional unit of the software system to fail in performing its required function; sometimes called a "bug", a fault is a part of *the code that needs to be fixed.*

3 ***Defect***

An anomaly in any intermediate or final software product resulting from an error or fault, ranging from an incorrectly specified set of test data to an incorrect entry in user documentation.

4 ***Failure***

Inability of a functional unit of the system depending on the software to perform its required function, or to perform the function within required limits.

Pfleeger (1992) classifies **errors, faults** and **defects** as the ***causes*** of the problem and **failures** as the ***effect*** of the problem. She emphasizes that the root cause of each problem needs to be determined in order to be able to assess its impact on software reliability.

5.2.2.2 Time

Software reliability metrics are usually defined within a time framework. Three "kinds" of time are usually involved: execution time, calender time and clock time. They are defined by Musa et al. (1990: 8) as follows:

Execution time for a program is the time that is actually spent by a processor in executing the instructions of that program.

Calender time is self-explanatory.

Clock time *represents the elapsed time from start to end of program execution on a running computer. It includes wait time and the execution time of other programs. Periods during which the computer is shut down are not counted.*

Failure occurrences in time are generally characterized (Musa et al. 1990: 9) as the

1) *time of failure*
2) *time interval between failures*
3) *cumulative failures experienced up to a given time, and*
4) *failures experienced in a time interval.*

5.2.2.3 The mean value function, the failure intensity function and the mean time to failure measure (MTTF)

Musa et al. (1990: 11, 18) define the above mentioned measures as follows:

*The **mean value function** represents the average cumulative failures associated with each time point.*

*The **failure intensity function** is the rate of change of the mean value function or the number of failures per unit time.*

*The **mean time to failure (MTTF)** is the average value of the next failure interval.* As this measure can be undefined, failure intensity is usually preferred as it always exists.

5.2.2.4 Availability

It is the expected fraction of time during which a software component or system is functioning acceptably. Availability is usually computed as the "ratio of up time to the sum of up time plus down time ... The down time is the product of the failure intensity and the mean time to repair (MTTR) ... **MTTR** is the average time required to restore the data base for a program, reload the program, and resume execution" (Musa et al. 1990: 18).

5.2.2.5 Software reliability

Musa et al. (1990: 15) define **software reliability** as: ***The probability of failure-free operation of a computer program for a specified time in a specified environment.***

An extension to this definition is given by Sheldon et al. (1992: 15) who defines **software reliability** as ***the probability of failure-free operation for a specified time in a specified environment for an intended purpose.***

The ANSI/IEEE (Standard 982.2) definition is: ***Software reliability is the probability that software will not cause the failure of a system for a specified time under specified conditions*** (Pfleeger 1992: 57).

Pfleeger (1992) discusses the difficulty in measuring reliability as defined by the IEEE/ANSI. Pfleeger points out that the software needs to be fully operational before reliability can be measured in this way. As it is far more cost-effective to solve problems while the software is being written, measures of reliability is also required in the development stage.

The above definitions represent a user view of reliability.

5.2.3 SOFTWARE RELIABILITY MEASURES

Software reliability measures are an effective means of determining and delivering the level of quality that the customer requires (Sheldon et al. 1992).

Figure 5.1 (Sheldon et al. 1992: 14) provides a view of the place of reliability measurement and modelling in the software life-cycle.

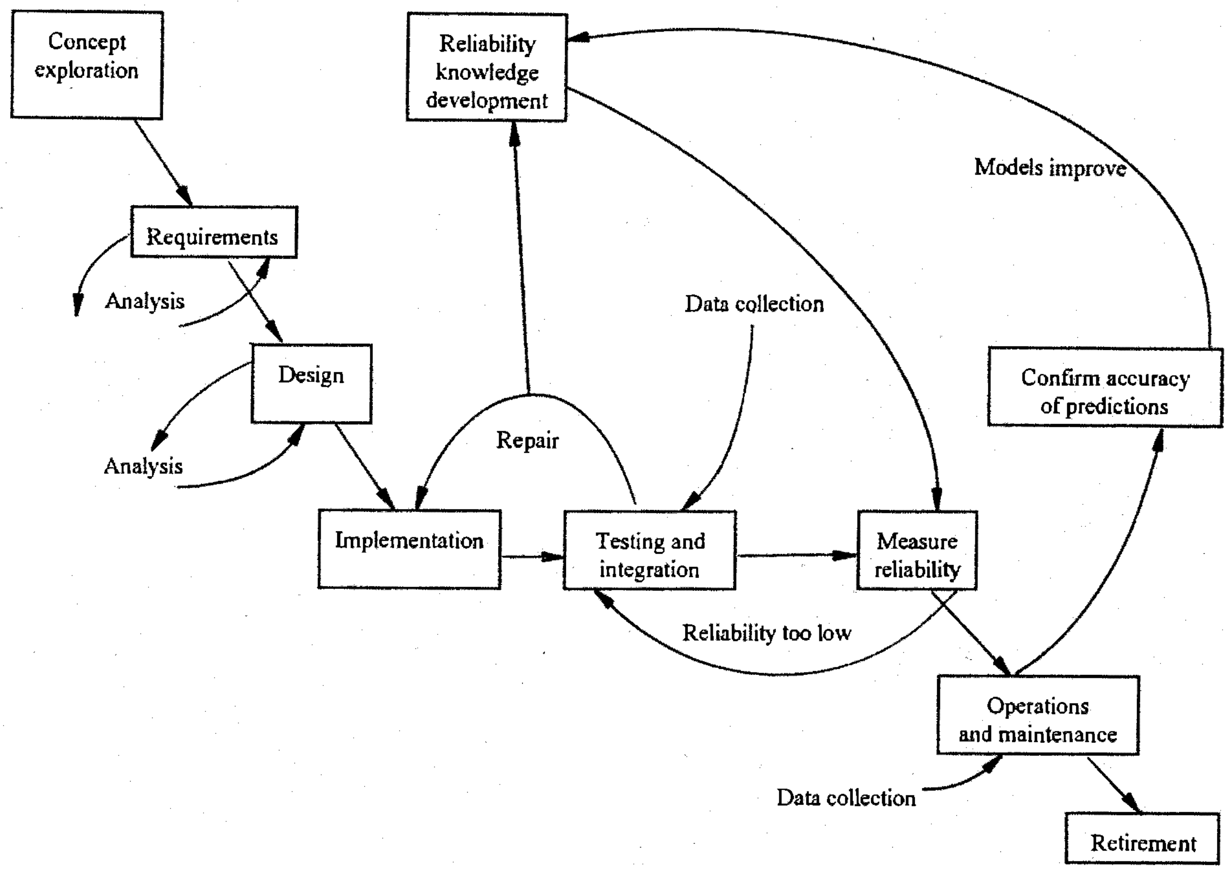


Figure 5.1 Reliability measurement and modelling in the software life-cycle

Useful reliability metrics can only be obtained by establishing the factors that influence the system's reliability. In addition, it is important to establish what constitutes a failure from the customer viewpoint. An operational profile also needs to be identified. This can be achieved by *gathering information on how previous versions were used, estimating the use of new features, and verifying the resulted estimated profile with the customer* (Sheldon et al. 1992: 15). This profile aids in planning test cases and data collection.

Everett, in Sheldon et al. (1992), remarks that the **number of faults** or **faults per thousand lines of code** is currently used as measures for software reliability in software development. He maintains that these measures are not good enough from the perspective of the customer. The customer's concern is **failures.** The frequency with which failures occur and their impact on business are important determinants of the customer's perspective of the quality of the software.

This view is also shared by Pfleeger (1992), who remarks that faults and defects are the aspects the software developer concerns himself with while the user is concerned about failures. Errors (as defined above) can be made by both developer and user. Examples of software reliability measures that are used in practise will now be described.

Fault density is one approach that is used by developers to measure reliability of finished code (Pfleeger 1992). Fault density is obtained by dividing the number of faults by the total number of lines of code in the final product. The number of faults is determined by tracking the total number of unique faults in a given time interval. The approach is used to judge testing thoroughness as well as to infer the operational reliability of the software.

Another approach is called **fault-seeding.** It is used to *estimate the number of faults remaining in the existing software.* Seeding take place by deliberately inserting faults into the software that are representative of the type of faults that have occurred in the past in similar projects. Fault searching takes place and the *ratio of discovered non-seeded to seeded faults* found is taken to indicate the number of faults remaining in the code. This approach is also used as a measure of test thoroughness and indirectly, of the reliability of the system (Pfleeger 1992: 57).

Pfleeger (1992) criticises the above-mentioned approach, as it does not look at failures in any specific context. She mentions the use of the technique of **failure profiles**. Failures are classified in categories in terms of the severity of their effect(s) on the system. The cumulative failures can then be tracked over time. It is possible to view the failure profile for the total system or for any part of the system. The technique can be used to project the completion of testing, assuming that there were sufficient test coverage.

Defect classification, another approach, helps in identifying the effect of defects on the reliability of the system. Defects are grouped in classes in terms of cause. It also aids in determining weights according to the criticality or severity of defects and to see where defects are introduced in the life-cycle (Sheldon et al. 1992).

Once measures for reliability are established, it is necessary to determine how these can be applied to control and ultimately improve software reliability (Everett, in Sheldon et al. (1992)).

Uses of software reliability measures include:

1) the quantitative evaluation of software engineering technology;
2) the evaluation *of development status during the test phases of a project*;
3) the monitoring of the operational performance of software and to control new features added and design changes made to the software (Musa et al. 1990: 21).

Everett, in Sheldon et al. (1992), lists the following uses of software reliability measures in practice:

1) The monitoring of the progress of system tests.
2) The prediction of the elapsed system test time in order to achieve a certain reliability objective.
3) The use of the reliability measures to change testing environments.
4) The exploration of how reliability measures can be used during development testing.

5.2.4 SOFTWARE RELIABILITY MODELLING

Sheldon et al. (1992: 15) define three broad stages of reliability modelling, i.e.

1) **assessment** (assumptions that are made regarding the environmental conditions under which the software will run)
2) **model development** (the derivation of mathematical expressions to estimate parameters such as failure intensity and the estimation of these parameters from real data through the use of statistical techniques) and
3) **measurement and estimation** (the use of the results to predict the behaviour of the software and to aid in planning and maintaining the software).

Sheldon et al. (1992) make a further important distinction between reliability prediction on the one hand, and reliability measurement (and estimation) on the other hand. Reliability prediction is based on static metrics (such as size and complexity) while reliability measurement (and estimation) is based on the dynamic execution of the program, e.g. failure data is collected during the system test.

Modelling of software reliability has mainly been focused on modelling reliability growth. Numerous models have been suggested (Musa et al. 1990). Two well known models that are discussed in Musa et al. (1990) are the basic execution time model and the logarithmic Poisson execution time model.

The software reliability model specifies the general form of the dependence of the failure process on the principal factors that affect software reliability and is time-based. These factors are fault introduction, fault removal and the environment (Musa et al. 1990). Sheldon et al. (1992) define the three principal factors that affect failure behaviour slightly different as: the number of defects (faults); the test strategy and operational profile; defect detection, removal and possible reintroduction. Furthermore, software reliability models are generally based on a *stable program executing in a constant environment* (Musa et al. 1990: 20). The models thus focus mainly on fault removal.

Most failure processes in software are random processes that vary with time. This type of process is called nonhomogeneous. The failure process is directly dependant on the environment or operational profile for the program. The **operational profile** of the program is defined ***as the set of run types that the program can execute along with the probabilities with which they will occur.*** A run is usually associated with some function that the software will perform. Runs that are identical repetitions of each other form a run type (Musa et al. 1990: 14).

The inverse relationship between failure intensity and the expected cumulative number of failures is the basis for most reliability models (Sheldon et al. 1992). The models differ *in general terms by the probability distribution of failure times or number of failures experienced and by the nature of the variation of the random process with time* (Musa et al. 1990: 19).

By plotting the observed failure rate as a function of the cumulative execution time, a reliability model can be statistically fitted to the data points. The fitted failure-intensity curve can then be used to estimate failure intensity and the additional execution time required to attain the failure-intensity objective. The failure intensity objective determines when software can be released.

Different characteristics of the failure process can be determined. According to Musa (1990: 19), analytical expressions exist for most models for:

1) *the average number of failures experienced at any point in time,*
2) *the average number of failures in a time interval,*
3) *the failure intensity at any point in time,*
4) *the probability distribution of failure intervals.*

5.2.4.1 Current modelling issues of interest

1) Yu, in Sheldon et al. (1992), indicates that software reliability models have little use in the testing environment as serious failures occur seldom in testing. He suggests that the customer's environment should be simulated to address this problem.

2) Everett, in Sheldon et al. (1992: 17), also raises the same aspect when he mentions the challenge they have faced practically, of modelling *how customers use software and how to set up appropriate test environments* in order to be able to apply the theoretical reliability models.

3) Current controversy on software reliability modelling is mentioned by Everett, in Sheldon et al (1992: 17), as: *which model is better, how well do the models reflect reality and how well do the models predict reality.*

4) The application of models that do not assume growth in reliability is considered as an avenue that needs further experimentation by Bazzana et al. (1993).

5) Another valid concern is regarding the collecting of "good data" (Brettschneider, in Sheldon et al. (1992)). He maintains that the collection of good data is the most difficult challenge in applying the reliability modelling theory. The criticality of complete, accurate and on-time data has been raised in the previous chapters and surfaces again when the

author discusses software cost estimation. The process of collecting data needs to be given a much higher priority and an infrastructure needs to be established within teams to achieve the aim of good data. The notion of "quality data" should receive more attention as the data forms the basis of all decisions.

The words of Brettschneider, in Pfleeger (1992: 60), reflects the critical role of software reliability measurement and modelling: *While measurement cannot ensure reliability, it can guide the development process and minimize the probability of unreliable software.*

5.3 SOFTWARE COST ESTIMATION

5.3.1 INTRODUCTION

The aim of software development companies is to produce cost effective quality software and to establish themselves as market leaders in their field. A successful software development project is therefore one that meets its cost, schedule and quality goals.

An internationally recognised problem in software organisations is overrun in terms of budget and time schedules. Manpower and elapsed time are considered to be the key costs in software development projects. Cost Estimation, defined as the empirical process of estimating effort and duration, and thus costs, is a serious problem for project management and has to be addressed. Lee, Lu and Lin (1994) list three aspects regarding software development that deems accurate software cost estimation a very difficult process: the unique requirements of each software project; the uncertainty involved in estimating the size of the software and the uncertainty of the user requirements. Improved effectiveness of both effort and duration estimation of software projects is therefore extremely important. The value of initial software estimates is totally determined by the amount of planning on which they are based. The ultimate aim is to develop an in-house process for a company that will provide accurate cost estimates. This will, in turn, improve the competitive position of the company.

Software cost estimation will be discussed as follows:

1) a strategic approach to software cost estimation
2) definitions of software cost estimation metrics
3) requirements for software cost estimation
4) software cost estimation models
5) the development of a local cost estimation model
6) software cost estimation tools
7) the use of a total installed cost template.

5.3.2 A STRATEGIC APPROACH TO SOFTWARE COST ESTIMATION

A strategic approach to software cost estimation is proposed, i.e. not prescribing the use of one technique or tool but recommending solutions for different aspects of the problem. Training, supported implementation, multiple estimation techniques and software cost estimation models as well as ongoing modification to the software cost estimation models are the core aspects of strategic application of software cost estimation in industry (Goodman 1992). Training and supported implementation were discussed within the context of a software metrics program in chapter 4.

Arifoglu (1993) proposes an integrated and general cost estimation methodology that supports the above strategy. The methodology suggests the use of a set of cost estimation methods to be applied step by step and in an integrated way to achieve improved results for planning and scheduling of a project.

The steps of his methodology (Arifoglu 1993: 102) are:

Step 1: *Estimate size*
Step 2: *Estimate effort and time costs*
Step 3: *Distribute effort and time costs to the life cycle*
Step 4: *Normalize Costs to actual calender time*

The steps can be depicted as follows (Arifoglu 1993):

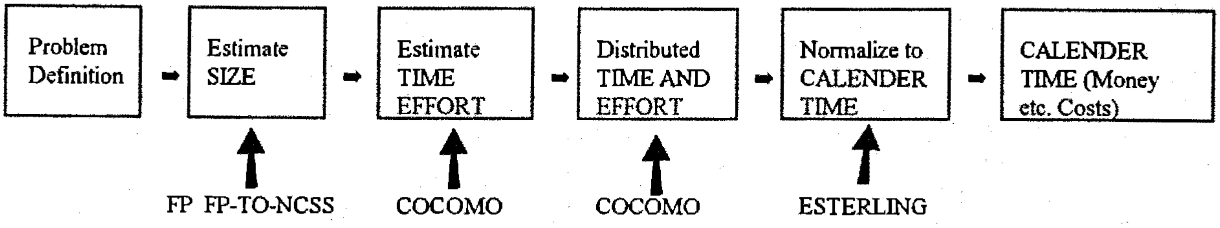

Problem Definition
Estimate SIZE
Estimate TIME EFFORT
Distributed TIME AND EFFORT
Normalize to CALENDER TIME
CALENDER TIME (Money etc. Costs)
FP FP-TO-NCSS
COCOMO
COCOMO
ESTERLING

Figure 5.2 The cost estimation methodology

where FP is function points, NCSS is noncommented source statements, COCOMO is the Constructive Cost Model (Boehm 1981) and Esterling is the Esterling model described in Esterling (1980). Function points and noncommented source statements will be defined in 5.3.3 and the COCOMO model will be discussed in 5.3.5.

Arifoglu (1993) suggests that, after these four basic steps have been performed, project management packages can be used for managing and scheduling of the project. The relationship between software cost estimation models and project management techniques is studied in 5.5.

Heemstra (1992) has also suggested the use of a cascade of software cost estimation models and techniques during the duration of the project instead of only one model or technique. As the project progresses, more reliable and accurate information becomes available which could be fully utilized through this approach.

The author is thus looking at software cost estimation from the broader perspective, i.e. the use and application of techniques and models within the framework of software quality management.

5.3.3 SOFTWARE COST ESTIMATION METRICS

The use of software metrics as a strategic tool, to improve the software development process, is internationally recognised and was discussed in chapter 4. Measurement and record keeping through a metric function will lead to better estimation, and thus control, of all projects (DeMarco 1982).

The concept of software metrics was defined in chapter 4.2.

The Goal/Question/Metric approach (i.e. the identification of measurement goals and important characteristics to be measured before defining the metrics) is widely used for determining the appropriate software metrics and ensures that they are defined with their intended use in mind. The Goal/Question/Metric approach was briefly described in chapter 4.3.2.

Software metrics especially designed for the object-oriented approach used in software development projects, is currently being developed in the literature (Chidamer & Kemerer 1991; Lorenz & Kidd 1994). The object-oriented approach comprises the modelling of the world or real-life situation in terms of objects and relationships between objects. An **object** is ***an abstraction of something in the domain of a problem or its implementation, reflecting the capabilities of a system to keep information about it, interact with it, or both; an encapsulation of Attribute values and their exclusive Services.*** A **class** is ***a description of one or more Objects, describable with a uniform set of Attributes or Services*** (Coad & Yourdon 1991: 4). It can also describe how to create new objects in the class. The primary motivation, according to Coad and Yourdon (1991: 5), is *to match the technical representation of a system more closely to a conceptual view of a problem domain and its implementation domain.* Booch (1991) identifies the following major principles of the object-oriented approach: data abstraction, encapsulation, modularity, inheritance, classification and polymorphism. A distinction is also made between object-oriented design (OOD), object-oriented analysis (OOA) and object-oriented programming (OOP). Metrics that have not been designed initially for this approach should be carefully assessed for appropriateness and validity within this environment, specifically in relation to cost estimation.

Desirable qualities for software metrics related to software cost estimation include:

1) early availability in project life cycle
2) the ability to standardise the metrics

3) high consistency in correlation to resultant cost and effort

4) acceptability to project personnel (DeMarco 1982).

The following software characteristics and associated metrics will be defined and briefly discussed as they play a crucial role in software cost estimation.

1 PRODUCT SIZE

There are currently mainly three metrics in use for the measurement of product size, namely

(i) ***Lines of code:***

Noncommented source lines of code (NCSS) and *Thousands of delivered source instructions (KDSI)* are the most used lines of code measures.

A delivered source instruction is defined by Boehm (1981: 58-59) as follows:

Delivered: This term is generally meant to exclude nondelivered support software such as test drivers. However, if these are developed with the same care as delivered software, with their own reviews, test plans, documentation, etc. then they should be counted.

Source instruction: The term includes all program instructions created by project personnel and processed into machine code by some combination of preprocessors, compilers and assemblers. It exclude comments and unmodified utility software. It includes job control language, format statements and data declarations.

Lines of code, is the oldest metric in use for product size. However, much controversy exist as to what a line of code constitutes. No clear definition exists across the software community (Arifoglu 1993). Jones (1986) identifies eleven major variations of line counting methods. This inhibits the comparison between software cost estimation studies.

The following factors also have to be considered when using parametric software cost estimation models that use lines of code as a product size measure (Wellman 1993: 38):

1) Code size is becoming less relevant as a guide to model effort.
2) *Executable lines of code are not usually comparable in terms of development effort, with other codes such as data definition, comments, etc..*
3) *Counting delivered code takes no account of the actual developed lines of code.*
4) Code size only applies to a part of the software development effort. There are significant costs incurred in software development that cannot be reflected by measures of code size or productivity. This aspect is also mentioned in Matson, Barret and Mellichamp (1994) who states that coding only accounts for 10-15% of the total effort. Parametric software cost estimation models thus provide an estimate for only part of the total software cost.

Lorenz and Kidd (1994) criticises the lines of code measure because:

1) Lines of code is not consistent across languages and applications. The end-user function of a line of code in e.g. Smalltalk versus Assembly differs dramatically. The lines of code measure is thus language dependent (Matson et al. 1994).
2) Code complexity is not reflected, and therefore not taken into account.
3) Using lines of code as a productivity measure encourages larger code volumes instead of less code with more functionality.
4) It is not a good predictor of quality or progress as we do not know anything about reliability, performance etc. of the software.

Matson et al. (1994) also raise a concern regarding the dependability of the lines of code measure on data available from past, similar projects.

(ii) ***Function points:***

The function points approach was developed as an alternative measure to the lines of code measure for size (Albrecht & Gaffney 1983).

In order to determine function points, the software is described in terms of the five user functions (Heemstra 1992: 633):

the external input type

the external output type

the external enquiry type

the logical internal file type

the external interface file type

The definitions of the user functions, and the levels of complexity (simple, average and complex) pertaining to each function, are described in Albrecht and Gaffney (1983) .

For each of these five types the number of simple, average and complex occurrences that are expected in the software is estimated. The assessment of complexity is based on the number of logical file accesses and/or data items affected by each feature (Kitchenham 1992). By weighting each estimated number with an appropriate weight (depending on whether they are simple, average or complex) a new number is obtained, the unadjusted number of function points, also called raw function points. The raw function point-count (RFP) which is the sum of the raw function points, is an indication of the nominal size of the software. A table extracted from Arifoglu (1993: 98) provides the weights to be used when calculating raw function points.

COUNT	WEIGHT SIMPLE	WEIGHT AVERAGE	WEIGHT COMPLEX
EXTERNAL INPUT	3	4	6
EXTERNAL OUTPUT	4	5	7
LOGICAL INTERNAL FILES	7	10	15
EXTERNAL INTERFACES FILES	5	7	10
EXTERNAL ENQUIRIES	3	4	6

Table 5.1 Calculation of unadjusted function points

Conversion tables are currently available in commercial tools to convert function points to NCSS (noncommented source statements) or vice versa. The conversion is used when function points are used as the product size measure but the software cost estimation model, e.g. the COCOMO model, that is used requires NCSS as input. This process is known as "backfiring" (Jones 1993). The accuracy of the conversion is not high (the range is ± 20%). It is however, important as an aid in the quantifying process.

Function points is currently the most widely used metric in software. The International Function Point User Group (IFPUG), was, in 1993, the largest software measurement association in the United States of America (Jones 1993).

Feature Points

Feature points method is an extended version of function points. One additional parameter, the number of algorithms that will be included in the application, is included in the calculation (Jones 1991, 1993). Systems and embedded software that are high in algorithmic processing lead to a larger total of feature points than function points. For Management Information System (MIS) projects, function points and feature points *come within a few percentage points of producing the same result* (Jones 1993: 14).

The basic structure for feature point calculation is (Jones 1993: 14):

Number of Algorithms × *3* =
Number of Inputs × *4* =
Number of Outputs × *5* =
Number of Inquiries × *4* =
Number of Logical Files × *7* =
Number of Interfaces × *7* =

The results can then be adjusted for complexity based on factors known as Problem Complexity and Data Complexity. Jones (1991) provides a detailed description of counting with feature points.

(iii) ***Object and method count*** (for object-oriented projects).

Object counts involved enumeration of all the object classes defined in the Objective C (or other language) programs that are developed.

Method counts tracked all operations defined on these classes (Pfleeger 1991a).

Several metrics, for application size, are suggested in Lorenz and Kidd (1994). Details can be found in Lorenz and Kidd (1994).

2 PRODUCTIVITY

Measures of productivity widely used are:

1) KDSI per person month (Thousands of delivered source instructions per person month).
2) Raw function points per hour.
3) A count of objects and methods per person month (for object oriented projects).

Factors that influence productivity in the specific environment of a company such as training; the amount of re-use; technology resources and experience (with domain/application)/(with development architecture)/(with tools/methods) need to be taken into consideration.

Books that concentrate on the aspect of productivity in the software industry and that can be consulted is *Software Engineering Productivity* (Stevenson 1995); *Software Productivity and Quality Today: The Worldwide Perspective* (Jones 1993); *Applied Software Measurement: Assuring Productivity and Quality* (Jones 1991) and *Programming Productivity* (Jones 1986).

3 EFFORT

The actual or reported person months of effort. A person-month is defined in South Africa as 160 working hours, i.e. 20 normal working days per month.

4 SCHEDULE/DURATION

A widely used metric for schedule/duration is:

Estimated progress, measured as *the ratio of the budgeted cost of the work done to the work scheduled.*

This metric uses standard cost reporting data on software work packages.

Another schedule metric defined by Möller and Paulish (1993: 72) are:

The difference between the planned and actual work time to achieve the milestone of first customer delivery divided by the planned work time.

It is indicated in percentages. A negative number will indicate a schedule slip.

5 QUALITY AND COMPLEXITY

Quality and complexity metrics are needed in the refinement stage of the software cost estimation modelling process. Only a few well known metrics are mentioned.

Quality

Widely used global metrics for quality are:

(i) **Pre-release defects** (Grady & Caswell 1987)

Grady and Caswell (1987: 56) distinguish between defects introduced, defects found and defects closed. They define ***defects*** as:

A defect is a deviation from the product specification or an error in the specification if the error could have been detected and would have been corrected. If the error could not possibly have been detected, or it could have been detected and would not have been corrected, then it is an enhancement, not a defect. Defects do not include typographical or grammatical errors in the engineering documentation.

Defects introduced: *The number of defects attributed to a flaw in the output of a particular activity which might not be found until a later activity. Do not include duplicates. (A duplicate refers to the case where the same defect causes more than one flaw.)*

Defects found: *The number of defects found in a particular activity. Do not include duplicates.*

Defects closed: *The number of defects corrected in a particular activity. Do not include duplicates.*

(ii) System test faults

The metric is obtained by *dividing the total number of software faults reported by the testing function during System Test by the number of thousands lines of code (KLOC) for each product for each release* (Möller & Paulish 1993: 69).

(iii) Customer change request

The metric is obtained by *dividing the number of unique change requests made by customers for the first year of field use of a given release by the number of thousand lines of code for that release. Only change requests which are faults detected by the customer will be counted. Feature enhancement change requests which are beyond the functionality documented in the software requirements specification are not counted* (Möller & Paulish 1993: 70).

Complexity

Complexity is defined as ***anything which increases the difficulty, and therefore the effort required by a programmer, to develop or maintain software*** (Conte et al, reported in Stevenson 1995: 265).

Two well known complexity metrics are the Halstead's E (and related T) and McCabe's v(G) (and the related DE) measures. The Halstead measure is a volume metric and is based on the number of operands and operators in a program (Stevenson 1995).

The McCabe measure is a *graphical ('cyclomatic') complexity measure which assumes that complexity depends on the decision structure (the number of paths) in a program, and not its size* (Stevenson 1995: 269).

5.3.4 SOFTWARE COST ESTIMATION REQUIREMENTS

Project Management typically requires the following from the software cost estimation process (Heemstra 1992):

1) How much time and effort will it cost to develop the software?
2) What are the dominating cost factors?
3) What are the important risk factors?

These questions are not easy to answer. Heemstra (1992) defines four core aspects that need to be taken into account when addressing the above questions:

A Reasons for problems associated with cost estimation.

B The prerequisites for estimation.

C The estimation of software development effort.

D Cost Estimation Models.

The first three aspects are discussed below. Cost Estimation Models will be discussed in 5.3.5.

A Reasons for problems associated with software cost estimation

Heemstra (1992: 628-629) lists the following reasons:

1) The lack of data on completed software projects (The importance of data collection was discussed in 4.3.4).
2) Estimates are often made in a hurry as estimators are being pressurised to write an estimate before the existence of clear specifications of the requirements of the system.
3) Specifications that are clear, complete and reliable are difficult to formulate at the start of a project. Adaptions and changes take place, therefore the budget also needs changing.
4) The characteristics of software and the development thereof, make estimating difficult.
5) The factors that have an influence on the effort and time to develop software, called "cost drivers". In practise, these cost drivers are difficult to determine.
6) Ongoing, rapid changes in information technology and software development

methodology are a problem for the stabilisation of the stimulation process, e.g. it is difficult to predict the influence of different prototyping strategies.

7) Experience in developing estimates is not common, especially for large software projects.

8) Software developers tend to underestimate effort.

9) The estimator tends to estimate the time it would take to perform the task personally. However, work will be done by different people with varying experience.

10) *There exists a serious mis-assumption of a linear relation between the required capacity per unit of time and the available time.*

11) In order to make a tender more acceptable, the estimator usually reduces the estimates marginally.

B The prerequisites for software cost estimation are:

(1) *Insight in the characteristics of the product that must be developed, the production means, the production personnel, the organisation of the production and the user/user organisation* (Heemstra 1992: 629). The above constitutes the cost factors. It is important for an organisation to consider the most dominant cost factors in its own environment.

When estimating, it is necessary to know *which cost drivers are the most important in the specific situation, what the values are of the drivers, and what the influences are on effort and duration* (Heemstra 1992: 629). In order to answer the above questions, the following issues regarding the cost drivers need to be addressed: definitions, quantification, correlation with other drivers, relationship between driver and effort, calibration, effectivity and efficiency, human factors and re-use (Heemstra 1992).

(2) Availability of a set of estimation models and techniques (Heemstra 1992).

(3) DATA - THE MISSING LINK TO SUCCESSFUL ESTIMATION

A critical requirement and thus prerequisite for software cost estimation that is often overlooked is the availability of good, reliable data. It is the author's view that the lack of data is the most pertinent constraint, identified to date, for successful software cost estimation.

The goal of successful software cost estimation can only be achieved if accurate, complete and on time data is available on projects. Data collection is the most important part of the process. It is not a trivial or free process. Without the necessary investment in data collection, no better estimates can be made using cost estimation models or methodology than can be done with a pure guesstimate.

The solution lies in a pre-operational investment in data collection that will ensure compliance to the data strategy. The strategy requires that

1) a structure for data collection must be set up which map with the cost drivers of the model to be used and
2) a mechanism must be established for maintaining the database and making it readily accessible to potential users.

Data availability and analysis also lead to identification of problem areas and is a definite value added function. As more data sets become available, they are used to check and improve initial norms. However, it must be kept in mind that the aim of the data collection is not the data itself, but the alignment of business goals that need to be achieved.

C The estimation of software development effort

In software development, the word "estimation" is used in the broader sense, as non-mathematical ways of estimation are included.

An estimation method is defined as successful in software development when it is easy to understand, refinable during the development process and the early estimation of the cost is within 25% of the actual final cost at least 75% of the time.

The primary estimation techniques used for software cost estimation (Heemstra 1992) are:

1. **Expert estimation**

   This type of estimation relies on an "expert" and its reliability depends on the ability of the expert to recall facts regarding a similar, completed project. The estimates are mostly subjective.

2. **Estimates based on reasoning by analogy**

   Information, through the existence of a database on previous similar projects, need to be available in order to use this technique.

3. **Estimates based on Price to Win**

   This cannot really be called "estimation". Commercial reasons are the only factor that influence the "estimate".

4. **Estimates based on available capacity**

   The basis of this method is the availability of means, especially personnel. Heemstra (1992) mentions one negative side effect, namely, that in the case of overestimation the planned effort will be used completely, an effect based on Parkinson's law: "Work expands to fill the available volume".

5. **Estimates based on the use of parametric models**

   The development effort and time are estimated as a function of a number of variables, the variables being the most important cost drivers. Parametric models will be discussed in 5.3.5.

The results of a survey mentioned in Heemstra (1992) indicate that the analogy method is mostly used (60,8%), but as it was found that only 50% of the organisations record data from completed projects, it is concluded that they worked on an informal analogy basis and not through the use of a database on historical projects.

In practise, a combination of these techniques is usually used. However, confusion exists in project manager's minds as to what "estimation" means. This was emphasized in an article by Edwards and Moores (1994) when they discuss the conflict between estimating and planning tools. This aspect will be discussed in 5.5.

Two main approaches to estimation (Heemstra 1992) can further be distinguished:

A **Top-down** approach where estimation is derived from global characteristics of the product and then split between the various components.

A **Bottom-up** approach where the cost of each individual component is estimated by the person responsible to develop the component. The costs are then added up to get the overall cost estimate of the project.

Arifoglu (1993) provides the categorisation of current approaches to cost estimation [reported in Shooman (1983)], namely Unit Cost or Price (estimate the cost for each sub-unit - the bottom-up approach); Percentage of Total Cost (estimate the software development component of the total system); Specific Analogy (using experience on previous, similar project to do estimation) and Parametric Equations (apply statistical techniques to historical data to obtain estimates).

5.3.5 SOFTWARE COST ESTIMATION MODELS

Software cost estimation models usually involve estimating the effort and duration of a software development project. It is mostly aimed at the macro level and is not specifically task-oriented.

Software cost estimation models have been developed since the mid-1960's. Statistical techniques such as regression and correlation are used to build the models based on measurements taken from software projects. The need for adjusting models due to the influence of cost drivers (factors that are perceived by project managers to have an important impact on costs) was recognised and cost drivers were built into the models from the mid-1970's.

Most software cost estimation models are "two-stage models". The first stage is a "sizer" and the second stage provides a productivity adjustment factor (Heemstra 1992: 631). An estimate of the size of the product needs to be obtained in the first stage. Metrics that have been used are lines of code and function points, and recently the use of object and method counts for object-oriented development (Pfleeger 1991a). The second stage provides an answer regarding the time and effort it will take to develop the software, usually in nominal man-months of effort, through the answer in the first stage. At this stage, factors known to influence the product at hand, the so-called cost factors, can be added to the model as the nominal effort does not take advantage of additional knowledge pertaining to the development. Application of this correction factor, often called a productivity-adjustment factor, provides a more realistic estimate.

The requirements for a Software Cost Estimation Model, provided by Heemstra (1992: 636) are:

A: MODEL REQUIREMENTS

It needs to be

- *Linked to the software control method*
- *Applicable at the start of a project*
- *Able to fit with the data that is available during development*
- *Possible to adjust estimate due to changing objectives*

B: APPLICATION REQUIREMENTS

- *Possibilities for calibration*
- *Accuracy of the estimates*

C: IMPLEMENTATION REQUIREMENTS

- *User-friendliness of the tool*
- *Possibilities for sensitivity analyses*
- *Possibilities for risk analysis*
- *Clarity of input definition*
- *Completeness and detail of output*

A general cost estimation structure (extracted from Heemstra 1992: 632) are depicted in figure 5.3:

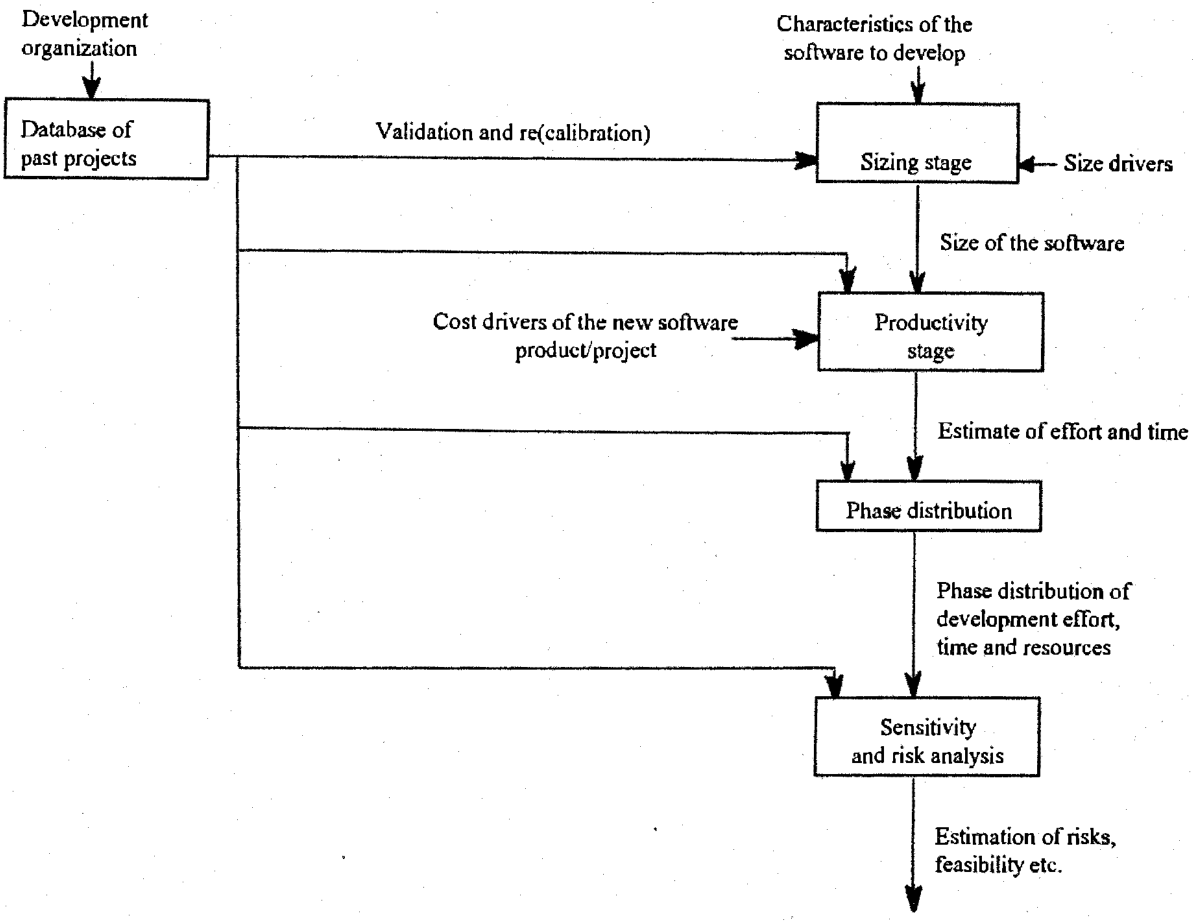

Development organization
Database of past projects
Validation and re(calibration)
Characteristics of the software to develop
Sizing stage
Size drivers
Size of the software
Cost drivers of the new software product/project
Productivity stage
Estimate of effort and time
Phase distribution
Phase distribution of development effort, time and resources
Sensitivity and risk analysis
Estimation of risks, feasibility etc.

Figure 5.3 A general cost estimation structure

Data on historical projects are the **critical input** to all aspects of the structure.

5.3.5.1 An overview of selected software cost estimation models

Well-known software cost estimation models are Boehm's COCOMO (Constructive Cost Model), and several extensions to it (e.g. GECOMO, TUCOMO and SECOMO), FPA (Function Point Analysis), Bailey and Basili's Meta model, and SPQR (Software Productivity, Quality and Reliability model). An extensive list of models and tools can be found in Heemstra (1992).

The software cost estimation models, COCOMO (Boehm 1981) and FPA (Albrecht & Gaffrey 1983) are representative of the two principal cost estimation directions, i.e. the use of lines of code versus function points as the size measure. Pfleeger (1991a) developed a model for software effort and productivity particularly useful when applied to object-oriented development and to assess re-use. An attempt to estimate and predict development effort of multimedia courseware using the Rayleigh curve (Marshall et al. 1994) is considered to be of importance with regards to the current explosion of multimedia activity. In addition, the development of a local cost estimation model with the help of a tool such as MERMAID MARK 1P is advocated (Kitchenham 1992). The above mentioned models will be described and discussed in terms of practical implications and shortcomings.

It is of paramount importance to stress that these models can only be useful if there is a good parameter database, on relevant past projects, available. Poor results in applying these models are primarily due to using these models incorrectly, e.g. without specific organisational data on past projects. Models are usually used without any calibration. The majority of available models also do not support calibration (Heemstra 1992).

1 The COnstructive COst MOdel (COCOMO)

The COCOMO model is the most well-known, well documented and transparent parametric cost estimation model (Heemstra 1992).

Three versions of the model exist: The basic model, the intermediate model and the detailed model. Basic COCOMO is for use when the projects is small to medium size projects in a familiar in-house software development environment. It gives a "quick, early, rough order of magnitude estimate" but has limited accuracy. The intermediate version includes cost factors in terms of their aggregate impact on overall project cost. Tables for the apportioning of the adjusted estimated effort and development time over the project phases exist. The detailed version of the model provides for the refinement of the adjustments for each phase. A set of phase sensitive effort multipliers for each cost driver attribute (to determine the amount of effort required to complete each phase) and a three level product hierarchy (the module, subsystem and system levels) are available in the detailed model (Boehm 1981).

The COCOMO model does not support the estimation of the size of the software. An estimate of the size (in KDSI) is essential for the use of the model.

Boehm (1981) provides a set of equations that represents the relation between size and effort and between effort and development time. The equations are based on 63 completed projects at the TRW company and were developed using a combination of analytical equations, statistical data fitting and expert judgement. The equations are listed in table 5.2 below (extracted from Heemstra (1992: 632) and Arifoglu (1993: 99)):

MODEL	DEVELOPMENT MODE	MAN-MONTH (NOMINAL)	DEVELOPMENT TIME (NOMINAL)
BASIC	Organic	$2.4*\mathrm{KDSI}^{1.05}$	$2.5*\mathrm{MM(nom)}^{0.38}$
BASIC	Semi-detached	$3.O*\mathrm{KDSI}^{1.12}$	$2.5*\mathrm{MM(nom)}^{0.35}$
BASIC	Embedded	$3.6*\mathrm{KDSI}^{1.20}$	$2.5*\mathrm{MM(nom)}^{0.32}$
INTERMEDIATE	Organic	$3.2*\mathrm{KDSI}^{1.05}$	$2.5*\mathrm{MM(nom)}^{0.38}$
INTERMEDIATE	Semi-detached	$3.0*\mathrm{KDSI}^{1.12}$	$2.5*\mathrm{MM(nom)}^{0.35}$
INTERMEDIATE	Embedded	$2.8*\mathrm{KDSI}^{1.20}$	$2.5*\mathrm{MM(nom)}^{0.32}$

Table 5.2 Equations for the COCOMO model

MM indicates man-months (a COCOMO man-month consists of 152 hours of working time) and is the measure for effort.

nom stands for nominal

Size is measured in **KDSI**, the number of thousands of delivered source instructions.

The development mode can be classified as organic, semi-detached or embedded.

Organic mode implies a *stable development environment, less innovative, relatively small size development;* **embedded** mode implies *developing within tight constraints, innovative, complex, high volatility of requirements;* and the **semi-detached** mode implies a development between organic and embedded mode (Heemstra 1992: 632).

The basic model is thus of the form

$$Effort = a \times (size)^b \times m(X)$$

where the values of a and b depend on the version of the model (basic, intermediate or detailed) and the development mode (organic, semi-detached or embedded) used and $m(X)$ is a composite multiplier which depends on the fifteen main cost driver attributes.

The fifteen COCOMO cost drivers (factors that are believed to affect the amount of effort required to produce a product) and their adjustment factor values are listed below in table 5.3 (extracted from Heemstra 1992: 633):

	Value of the cost drivers					
Cost drivers	Very low	Low	Average	High	Very high	Extra high
Required reliability	0.75	0.88	1.00	1.15	1.40	
Database size		0.94	1.00	1.08	1.16	
Complexity software	0.70	0.85	1.00	1.15	1.30	1.65
Constraint execution time			1.00	1.11	1.30	1.66
Memory constraints			1.00	1.06	1.21	1.56
Hardware volatility		0.87	1.00	1.15	1.30	
Response time constraints		0.87	1.00	1.07	1.15	
Quality analysts	1.46	1.19	1.00	0.86	0.71	
Experience with application	1.29	1.13	1.00	0.91	0.82	
Quality programmers	1.42	1.17	1.00	0.86	0.70	
Hardware experience	1.21	1.10	1.00	0.90		
Programming language experience	1.14	1.07	1.00	0.95		
Use modern programming techniques	1.24	1.10	1.00	0.91	0.82	
Use software tools	1.24	1.10	1.00	0.91	0.83	
Project duration constraints	1.23	1.08	1.00	1.04	1.10	

Table 5.3 The COCOMO cost drivers and their influence on the nominal effort

The adjustments are multiplicative. An example of their use is: If the complexity of the software is high, the nominal effort needs to be multiplied by 1.15.

A comprehensive and detailed explanation of the COCOMO model is provided in Boehm's book: **Software Engineering Economics** (Boehm 1981). Recently, the use of the COCOMO model in object-oriented development (by adjusting cost factors to account for aspects of object-oriented development) was described in Pittman (1993).

2 Function Point Analysis (FPA):

Function point analysis was developed by Albrecht (1979) [reported in Albrecht & Gaffney 1983]. He developed function points as an alternative measure to the lines of code measure. The method is programming language or fourth generation tool independent. It is based on the number of "functions" that the software has to fulfil. These functions are related to the data the software uses and generates. The term "function points" was explained in 5.3.3.

The raw function points (RFP) can also be adjusted to provide the adjusted function points, by using the following 14 technical characteristics (Heemstra 1992: 634):

$$Adjusted\,function\,points = RFP \times TCF$$

where *TCF* (technology adjustment factor) is calculated as follows:

$$TCF = 0.65 + 0.01 \times Sum(DI_i)$$

and $Sum(DI_i\,)$ is the sum of the degree of influence rating for each of the 14 factors (characteristics) (Kitchenham 1992). The ratings are obtained by rating each characteristic in terms of the degree of influence.

The numbers 0 to 5 are used to indicate the degree of influence where

0: not present or no influence;
1: insignificant influence;
2: moderate influence;
3: average influence;

4: significant influence;

5: strong influence or essential.

The raw function point count can increase or decrease by a maximum of 35% by using TCF as a multiplicative adjustment factor (Kitchenham 1992).

A shortcoming of function point analysis is the fact that it has been developed for use with business applications and needs restructuring for use with real time and mathematical applications because of the totally different internal structure and complexity of these types of software (Wellman 1993). The shortcoming has, however, to a great extent been covered by feature points (Jones 1991).

Another disadvantage is the amount of subjectivity involved in calculating function points (MacDonell 1994). However, this aspect is common to all suggested models that involve the rating of cost factors. Kemerer and Porter (1992) have studied the reliability of function point measurement through an empirical study, specifically the inter-rater reliability of function point counts, i.e. whether two individuals performing a function point count for the same system would produce the same results. They conclude that generally function points is more reliable than what are casually believed but point out areas where improvements could be made.

Another version of function points, the Mark II Function Points, to improve certain weaknesses, have been suggested by Symons (1988).

3 Pfleeger's model of software effort and productivity (with specific application to object-oriented software development).

Pfleeger (1991a) describes a new model of software effort and productivity. A key characteristic of the model is the ability to *amortize the cost of a factor over the projects that may be affected by the factor* (Pfleeger 1991a: 224). It thus allows for the assessment of the cost of re-use. The outline and structure of the model will be briefly described.

Pfleeger (1991a: 224) defined general productivity as:

$$Productivity = \frac{Outputs\ produced\ by\ the\ process}{Inputs\ consumed\ by\ the\ process}$$

She regards the inputs to represent the amount of effort required to produce the final product and the outputs to represent the final product.

She thus rewrites the equation as

$$Effort = \frac{Product}{Productivity}$$

If organisations develop similar types of software and use similar type of techniques and methods in the development, Pfleeger (1991a) suggests that one can examine the average productivity and use it as predictor for productivity of a future project.

Pfleeger (1991a) defines:

$$P_{actual} = P_{avg} \times f$$

where P is productivity and f is a composite multiplier that adjust the average productivity to account for characteristics of a project.

The model involves six steps, detailed by Pfleeger (1991a: 224) as:

1) *determination of average productivity*
2) *identification of the major factors that affect productivity in a way different from the usual case*
3) *determination of the amount of the project affected by each factor*
4) *computation of the multiplier that captures the effects of each factor on the upcoming project*
5) *determination of the overlapping effects of combinations of factors*
6) *calculation of a composite multiplier to reflect the effects of all factors on the projects.*

A cost factor is defined as ***any aspect of the development process that can influence effort or productivity significantly*** (Pfleeger 1991a: 224).

The **cost multiplier** for cost factor X is defined as the

$$\frac{\textit{Effort needed for development using factor } X}{\textit{Effort needed for development without using factor } X}$$

Overall effort thus decreases if the above multiplier is less than 1.

The function f thus *represents the combination of one or more cost multipliers for factors that have a significant effect on the development project* (Pfleeger 1991a: 225).

Pfleeger (1991a: 225) derived the following equation for the *effort needed for development for a given project relative to the use of the cost factor X:*

A_x = (relative effort for part of project not affected by *X*) × (portion of project not affected by *X*)

+ (relative effort for reapplicable part of *X*) × (portion of project involving *X* that can be reapplied to other projects)

+ (relative effort for project-specific part of *X*) × (portion of project involving *X* that is project-specific).

i.e. $$A_x = 1(1 - D_x - G_x) + (K_x/m_x)G_x + (b_x + K_x/n_x)D_x$$
$$= 1 + D_x(b_x + K_x/n_x - 1) + G_x(K_x/m_x - 1)$$

where X is the cost factor that affect effort;

A_x represents the ratio of the effort during development using X to the effort during development without the use of X;

R_x represents the portion of a project affected by X;

b_x represents the effort on a project to incorporate X compared with the effort needed for the project if X was not used (it reflects the effect of X only on R_x);

K_x is the relative effort to create a factor X;

D_x indicates the portion of the project that involves X that can be applied to other projects;

G_x indicates the portion of the project that involves X that is project-specific;

$$D_x + G_x = R_x \, ;$$

m_x is the number of uses over which the reapplicable portion of X is to be amortized;

n_x is the number of projects for which X is specifically designed or designated.

Pfleeger (1991a) gives an equation for N_x, the minimum value of n_x for which the creation of X "pays off". It is the value of n_x for which the value of A_x will change from greater than 1 to less than 1.

$$N_x = \frac{K_x(G_x + D_x)}{G_x + D_x(1 - b_x)}$$

The project effort estimates can suggest *how often a cost factor should be re-used or reapplied to other projects to keep the overall per-project costs low* (Pfleeger 1991a: 225). Most known models use size to estimate the nominal effort. In the model suggested by Pfleeger (1991a: 226), *size is considered in the context of where and why additional effort is required.*

Pfleeger (1991a) combines the equations for the cost factors and generalizes it to the case of t cost factors. Details of the derivation can be found in Pfleeger (1991a).

The approach suggested by Pfleeger requires subjective judgement. In discussing ways to minimize subjectivity, Pfleeger (1991a) suggests the following:

1) In an object-oriented development, measure productivity as the count of objects and methods per person-month (the newer equivalent of man-months!) available at that stage (e.g. requirements, design, etc.) of the development. The counting can thus be made at the beginning of development and then again throughout the development process. This will

maximise the use of available information and minimize subjectivity.

2) She particularly emphasized the important role that a database of projects and organisational characteristics, an aspect continuously stressed by authors on software metric programs (chapter 4) and software cost estimation, can play in this regard. The use of such a database will, in the long run, minimize the subjectivity of the estimate as the choice of factors and values for b, K, n, m, G and D requires knowledge and understanding of previous projects and development environments.

Pfleeger (1991a) made a preliminary empirical comparison between her model and COCOMO to determine whether her proposed model predicts effort and productivity more accurately than COCOMO. She stresses that other aspects, such as using the model as an instrument to make choices between alternative strategies or evaluating trade-offs have not been investigated yet. She used three software development projects that involve object-oriented methods and Objective C as the development language. She found her model to perform better than the COCOMO model, but add that additional research with larger datasets is necessary to validate the model.

As "reuse" is becoming more and more important in software development, the model suggested by Pfleeger, which reflects reuse of any aspect, has a definite application in software cost estimation.

4 A Composite Model for Development Effort of Multimedia Courseware

The development of multimedia courseware requires substantial effort. Marshall et al. (1994) proposes a model, MEEM (Multimedia Effort Estimation Method) to predict development effort of multimedia courseware.

A waterfall model of multimedia courseware development was proposed to aid in the development of appropriate metrics. The waterfall model is a commonly used phased based model for the software development life-cycle (Boehm 1981). Within this model, each phase (e.g. program coding) is well-defined with start and end-points (Marshall et al. 1994).

The basis for the proposed MEEM model is the COCOMO model. The MEEM model is defined as:

$$Effort = a \times (Average\ Training\ Delivery\ Hours)^b \times CD(X)$$

where effort is measured in person-hours; Average Training Delivery Hours is an initial estimate of the number of hours of training required and CD(X) is a cost driver that depends on the number of factors which affect the development of multimedia courseware. The values of constants a and b are used to *map data onto the proposed model and to convert average student hours into development staff hours* (Marshall et al. 1994: 253).

Marshall et al. (1994) list and discuss possible multimedia cost drivers. The cost drivers can be grouped into four categories: Course Difficulty (CD); Interactivity (IN); Development Environment (DE) and Subject Expertise (SE). The cost drivers are currently defined in terms of an ordinal scale (very low; low; normal; high and very high). Validation has to take place through experimental data and statistical analysis (Marshall et al. 1994).

Concerns regarding the model raised by Marshall et al. (1994) are:

1) the existing debate on the validity of using Average Training Delivery Hours as the basis for a metrics-based model,
2) the fact that the model is based on the assumption that staff utilization during development can be modelled as a Rayleigh curve and
3) the independence of the cost factors.

Marshall et al. (1994) analysed 14 courseware development projects. They studied the relationship between the groups of cost drivers and delivery/development time. The scores were obtained by adding the ratings for each cost driver within a group, thus assuming equal weight of the cost drivers within a group. They also assume that the scale, a set of ordered categories, may be approximated by an interval scale.

Marshall et al. (1994) admit shortcomings in their analysis due to the small data set. They stress that their current model is a framework and cannot be, as yet, used for estimation. Calibration with a large data set is necessary to determine the coefficients and cost driver values.

The attempt to address the estimation of multimedia courseware development effort by Marshall et al. (1994) is seen by the author as an important contribution to the aspect of software cost estimation modelling.

5 The Development of a Local (In-House) Software Cost Estimation Model

The development of a local software cost estimation model is suggested by the MERMAID project team who was appointed to develop and automate improved methods of cost estimation. The model is based on locally (in-house) collected data (Kitchenham 1992)

The author proposed the following flowchart to develop a local software cost estimation model for project planning in a software development company:

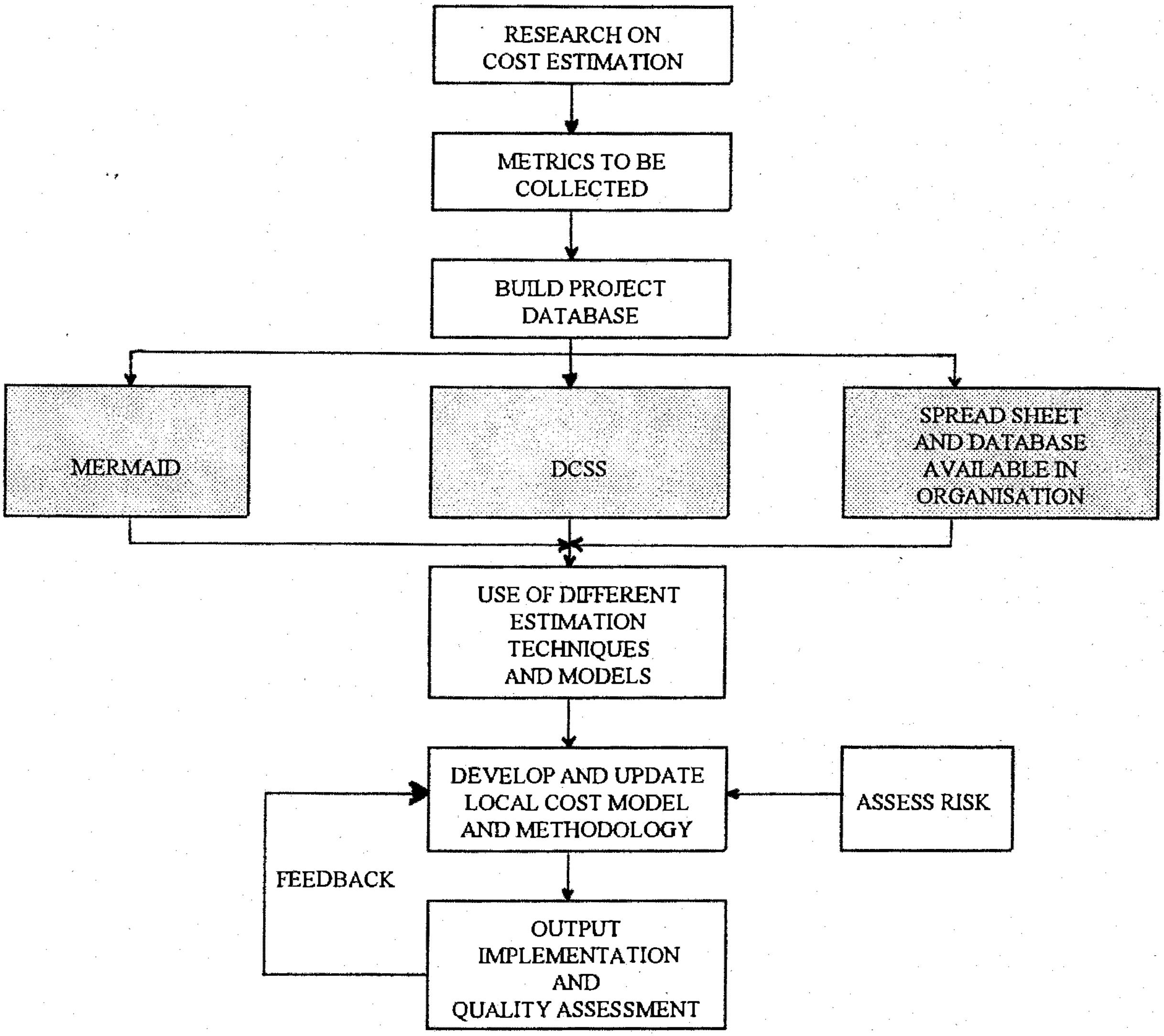

RESEARCH ON COST ESTIMATION
METRICS TO BE COLLECTED
BUILD PROJECT DATABASE
MERMAID
DCSS
SPREAD SHEET AND DATABASE AVAILABLE IN ORGANISATION
USE OF DIFFERENT ESTIMATION TECHNIQUES AND MODELS
DEVELOP AND UPDATE LOCAL COST MODEL AND METHODOLOGY
ASSESS RISK
FEEDBACK
OUTPUT IMPLEMENTATION AND QUALITY ASSESSMENT

Figure 5.4 The development of a local software cost estimation model

The establishment of a sound project history database is an essential prerequisite of a local cost estimation model. The definitions of the counting rules for, and measurement of, size, effort and duration are made and agreed upon by the organisation developing their own software cost estimation model. By using an in-house developed model, the use of subjectively determined adjustment factors can be minimised as only the cost drivers that are relevant to the software development company in question and the specific project environment are used. The model can be refined as more project data becomes available. The model can be updated using feedback from the assessment of the associated risks and the output and implementation stage.

The MERMAID MARK 1P tool, that will be described in 5.3.6, can be used to build a database of projects in order to construct a local cost estimation model.

5.3.5.2 Current software cost estimation models' issues

Certain aspects that are currently researched are:

1 **Validation**

Validation of software cost estimation models proved to be difficult because of the lack of data on completed software projects. Heemstra (1992: 635) remarks *data collection is not common in the software community. It is labour and time-intensive and requires an attitude not only focused on the constructive part but also on the analytical part of software engineering.*

2 **Accuracy**

Studies by Heemstra (1992) and Kemerer (1987) indicate that the different models' estimates vary considerably. An important aspect forthcoming from the studies, is the need for calibration of models within an organisation's own environment.

3 **Software Size Estimation**

Software cost estimation models usually require an estimate of the size of the software as input early in the project. This is by no means an easy task. The sizing question is usually addressed through the use of either the lines of code measure or the function point measure. It remains difficult as specific knowledge about the future system's complexity, interactions and scope is required.

Subjective techniques and objective models have been proposed in the literature (Laranjeira 1990). One of the most popular subjective techniques used is based on the PERT technique (Pressman 1993; Putnam & Fitzsimmons 1979). It utilizes the "expertise" of the personnel involved in making the estimate.

Three values for each measure, e.g. function points, are estimated: a pessimistic, optimistic and most likely value.

The expected estimation value are then given by

$$E = \frac{(a+4m+b)}{6}$$

where a is the optimistic, b is the pessimistic and m is the most likely value.

It will have an estimated standard deviation of

$$s = \frac{(b-a)}{6}$$

This will, in turn, lead to the observation that, provided no change occurs in the product requirement definition, the size of the model to be developed will have a probability of

99.8% of being between E-3s and E+3s
95% of being between E-2s and E+2s
68% of being between E-s and E+s

The PERT method and it's estimation of uncertainty will be discussed in chapter 6.

Laranjeira (1990) proposes a method for software size estimation based on an object-oriented specification model and on statistical methods. Other recent work include the Mark II Function points (Symons 1988); feature points (Jones 1991) and a software size model suggested by Verner and Tate (1992).

4 Independence, subjectivity and relevance of cost factors

Kitchenham (1992) studied the role of the productivity adjustment factors, called cost drivers. The main criticisms against using models with productivity adjustment factors centre around the independence of the factors, the applicability of these factors in all organisations and the subjective evaluation required in using the factors. The MERMAID project team (Kitchenham 1992) use analysis of variance to study the relationship between certain productivity factors and productivity (measured as size/effort). Two empirical data sets, the MERMAID-1 and MERMAID-2 data sets were used for the analysis. The

MERMAID-1 data set comprises data on 81 software development projects and the MERMAID-2 data set comprises data on 30 software development projects. They found statistical evidence that the use of fourth generation languages (4GL) are associated with improved productivity. General improvement in software engineering methods and tools does not show a significant relationship with improved productivity. Furthermore, after studying the effect of certain staff characteristics on productivity (using the original data set that was used to develop the COCOMO model), Kitchenham (1992) concludes that it seems that team differences cannot be measured such that it can be used to improve the precision of software cost estimation. A principal component analysis to test the assumption of independence of 21 available productivity adjustment factors (cost drivers) was also performed using the MERMAID-2 data set. It was found that seven principal components account for 76,2% of the variability in the data. This suggests that the cost drivers are not independent.

Kitchenham (1992) also investigates the effect of the use of adjusted function points on the relationship between size and effort. She shows that their results for the MERMAID-1 data set (by using the Pearson correlation coefficient) do not indicate a significant improvement in the relationship between size and effort if adjustment factors for size are used. She concludes that adjustment factors are not necessary in a single environment. In addition, analysis of variance carried out on the MERMAID-2 data set, indicate that only three of the adjustment factors ("data/control information sent/received over communication lines; online data entry and control functions; online update for internal files") were related to productivity. Kitchenham (1992) also tests the independence of the fourteen function-point adjustment factors through the use of principal component analysis on the MERMAID-2 dataset. It was found that six principal components accounted for 85,5% of the variability of the data and none of the remaining components accounted for more than 5% of the variability of the data. It thus indicates that the original factors are not independent.

5 The effect of schedule compression

Kitchenham (1992) also studied the effect of schedule compression on effort and productivity. The empirical results did not support any of the schedule compression models that are currently included in cost estimation models.

6 **The relevance of point estimates**

Heemstra (1992) remarks that project managers would rather want to have a number of scenarios from which alternatives can be chosen and would like to know the sensitivity of an estimation to specific cost drivers. By taking such an approach to estimation, more insight into the problem is gained. It would also provide a basis for project control.

7 **Definitions and standards**

An important prerequisite for successful estimation is the development, acceptance and use of a uniform set of definitions and standards (Heemstra 1992). He envisages a more structural approach to estimation.

8 **The use of a cascade of techniques**

This aspect has also been mentioned as part of the strategy for software cost estimation. Heemstra (1992: 638) remarks: *The lack of accurate and reliable estimation techniques combined with the financial, technical, organisational and social risks of software projects, require frequent re-estimation during the development of an application and the use of more than one estimation technique.*

9 **The non-linear relationship between development effort and software size**

The nonlinear relationship between development effort and software size is still an active area of debate. It will be discussed in 5.4.

5.3.6 SOFTWARE COST ESTIMATION TOOLS

Tools are required for the support of the collection and reporting of the metrics. The tools have to be selected to support the activities of the development process. In addition, tools must be flexible to allow for maintenance and updating as the process changes due to improvement.

The ability to migrate tools to an electronic platform and thus reduce effort and increase efficiency is required.

An ideal tool for a cost estimation model should support project management in the following seven steps (Heemstra 1992: 631):

1) *Creation of a database of completed projects*
2) *Size estimation*
3) *Productivity estimation*
4) *Phase distribution*
5) *Sensitivity and risk analysis*
6) *Validation*
7) *Calibration*

The aspects of calibration and sensitivity and risk analysis are usually lacking in the available tools (Heemstra 1992).

Edwards and Moores (1994) define a cost estimating tool as consisting of:

1) a mathematical model (M) which relates known properties of the system (K) to useful unknown properties (U), such as cost and duration
2) adjustment factors (A) which relates the generic model to a particular project
3) an interface (I) such that the user can determine the effect of K and A on U.

Several commercially available tools for software cost estimation exist. BYL, ESTIMACS, GECOMO, SLIM, SOFTCOST and SPQR/20 are some of the well-known estimating tools. Extensive lists of tools can be found in Heemstra (1992) and Hetzel (1993). Two newly developed tools, MERMAID MARK 1P and MEIS, a tool developed by Ariflugo (1993), will be described briefly.

1 MERMAID MARK 1P

The prototype MERMAID MARK 1P cost estimation tool is used for the establishment of an initial baseline for software cost estimation. The approach used is based on the collection of local (in-house) data and the generation of local cost estimation models from that data.

An evaluation copy of MERMAID MARK 1P was obtained from the National Computing Centre in Manchester, United Kingdom. The tool was developed as part of the research carried out for the MERMAID project (MERMAID MARK 1P ... 1992).

MERMAID MARK 1P provides the following features:

1) *Support for defining standard project lifecycles consisting of consecutive milestones with project attributes defined both at the project level and the milestone level - CONFIGURATION tool*
2) *Support for defining projects, based on such standard project lifecycles or fully free format, consisting of consecutive milestones and with project attributes defined both at the project and at the milestone level - PROJECT DEFINITION tool*
3) *Support for the upgrading of project structures to standard project lifecycles - PROJECT DEFINITION tool*
4) *Data entry of attribute values for individual projects - DATA ENTRY tool*
5) *Analysis of historic and present project data through tabular and graphical representation of the data, including value distribution plots, box plots, trend plots and scatter plots - ESTIMATION tool*
6) *Estimating effort, duration or other project attributes through statistical techniques, using historical projects as baselines for such estimations - ESTIMATION tool*
7) *Storage of estimates, whether made through the statistical analysis or by hand - ESTIMATION tool*
8) *Analysis of the estimates over time versus the actual in order to make a post mortem assessment of the estimation process - ESTIMATION QUALITY ASSESSMENT tool*

2 MEIS - Measurement and Evaluation Package

Arifoglu (1993) integrated his cost estimation methodology in a tool called MEIS (Measurement and Evaluation Package). It includes the automation of the methods of Function Points, FP-to-NCSS Conversion, COCOMO and Esterling. It is developed for a microcomputer environment.

5.3.7 A TOTAL INSTALLED COST TEMPLATE - THE ANSWER?

Software cost estimation models provide only a part of the total software cost.

A total installed cost template that aggregates costs from various sources, namely estimates for software development, quoted costs for third party software, project management, data transfer and training (Wellman 1993) should be the ultimate goal to strive for in the cost estimating process.

The template suggested by Wellman (1993: 46) is:

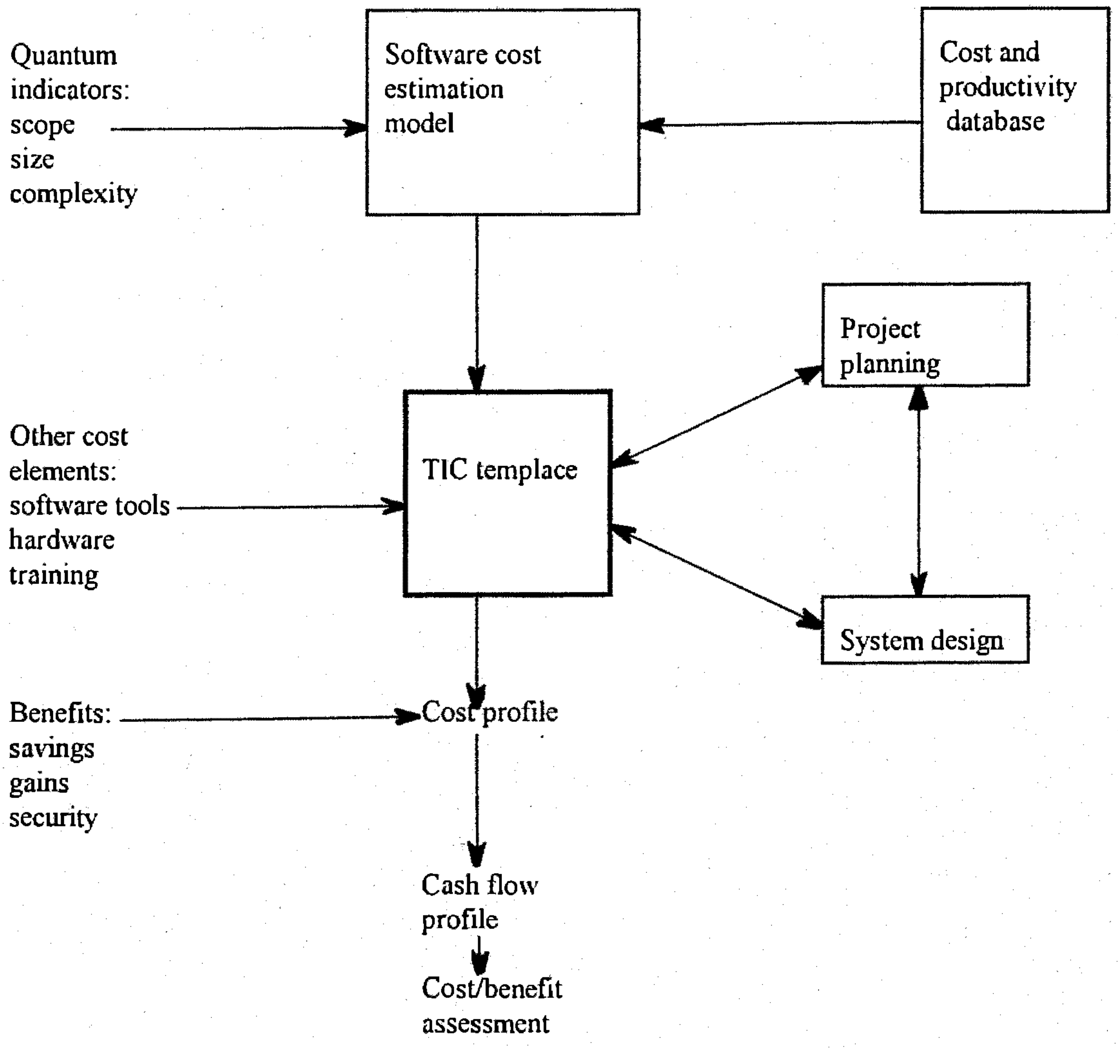


Figure 5.5 TIC template in the cost estimating process

5.3.8 CONCLUSION

Software cost estimation models are currently not generally accepted in industry. The incorrect use of the models is mainly responsible. As mentioned before, calibration is necessary for useful results.

Theory and practise have proved that no individual technique, metric or tool is ideal or universally applicable.

Good software cost estimation is ultimately based on the understanding and use of a range of tools and techniques and the judgement of an expert as to which combinations are the most appropriate in each situation.

The software cost estimation process is dynamic. As requirements change and more information becomes available, the model(s) used needs to be adapted.

The following aspects are paramount to the success of this iterative process:

1) Re-estimating throughout the life of a project. Continuous updating of product size, required effort, cost and schedule estimates are very important.
2) Using different techniques as independent checks. Techniques that can be used include the Delphi approach, a combined SSM/PERT technique, functional/structural decomposition, statistical analysis and estimation by analogy (Goodman 1992).
3) Comparing actual and estimated values. The output generated must be carefully compared to actual values and discussed with the development's project manager and team in order to make an assessment as to the quality of the estimate.

One of two approaches (or a hybrid of these) can be followed for optimal efficiency and effectivity when implementing the process of software cost estimation in an organisation, namely

1) Use an established model but calibrate it for the specific environment or
2) Develop a local (in-house) cost estimation model

Both of these cannot be established without a database of projects. Only when this has been accomplished, can the model be of quality usage.

The two prototype tools, MERMAID (a cost estimation tool) and DCSS (data collection and storage system tool), can be used for implementation of a software cost estimation process in a software development company.

5.4 SOFTWARE COST ESTIMATION MODELS - THE NONLINEAR QUESTION REVISITED

5.4.1 INTRODUCTION

Software cost estimation models are generally of the basic multiplicative form

$$y = a\,x^{b} \qquad (1)$$

where y indicates the development effort and x indicates the size of the software development project.

An analytical aspect of software cost estimation models is that they assume a non-linear relationship between size and effort (Kitchenham 1992).

Substantial debate has been going on regarding the validity of the nonlinear relationship. This aspect will now be discussed and analysed.

5.4.2 VIEWPOINTS ON THE NONLINEAR ASPECT OF SOFTWARE COST ESTIMATION MODELS

Kitchenham (1992) has challenged the assumption of nonlinearity. She tests the assumption empirically, using published software cost estimation datasets, as well as three datasets from the MERMAID-project.

By means of linear regression, applying the transformation

$$ln(y) = ln(a) + b\ ln\ (x) \tag{2}$$

she obtained estimated values of *b* for each dataset and tests whether this term differs significantly from *1*. Except for one case (a subset of one of the datasets), the datasets tested in the study do not have an exponential term that differs significantly from *1*. Kitchenham (1992) concludes that, within a single environment, the *b* term does not differ significantly from *1* and that a linear model, i.e. $y = ax$ is likely to be sufficient.

Banker et al. (1994) challenge Kitchenham's (1992) outcome. They investigate the aspect of nonlinearity in software development effort modelling, where software development is seen as an economic production process. Banker et al. (1994) use regression modelling as well as new semi-parametric statistical tests with the Data Envelopment Analysis methodology. The datasets used are all the datasets cited in Kitchenham (1992), except the MERMAID-3 dataset. In addition, they use two other datasets cited in Banker and Kemerer (1989).

Banker et al. (1994) indicate that the model:

$$ln(y) = \beta_0 + \beta_1 ln(x) \tag{3}$$

where y = effort, x = size, β_0 corresponds to *ln(a)*, and β_1 to *b*

does not allow for the possibility of both increasing and decreasing returns to scale.[1] The hypothesis of both increasing and decreasing returns to scale in the same dataset can thus not be tested with this model.

Banker et al. (1994) indicate that the problem with Kitchenham's analysis (Kitchenham 1992) is one of probably misspecifying the model and thus making inaccurate inferences about the β_1

[1] A production process exhibits local increasing returns to scale if, at a given volume level, the marginal returns of an additional unit of input exceed the average returns. Local economies of scale is thus present when average productivity is increasing, and scale diseconomies prevail when average productivity is decreasing (Banker et al. 1994: 275).

coefficient.

Banker et al. (1994) state that it is better to estimate a form of the production model that will allow for both economies and diseconomies of scale and perform the appropriate tests in that case. They suggest the quadratic model

$$y = \alpha_0 + \alpha_1 x + \alpha_2 x^2 \tag{4}$$

and test whether $\alpha_0 = 0$ and $\alpha_2 = 0$, arguing that these tests will give an indication of whether a linear relationship between size and effort is adequate.

The results indicate that the hypothesis ($\alpha_2 = 0$) is rejected, at the 5% level of significance, for 6 of the 11 datasets. The White-heteroscedasticity-consistent estimator of the covariance matrix was also applied to calculate the t-statistics (because assumptions required to apply the regression model seem to be violated). The null hypothesis of $\alpha_2 = 0$ was again rejected at the 5% level of significance in six of the eleven cases. Banker et al. (1994) consider this as rejection of a linear relationship between project size and effort. They also applied Fisher's exact chi-square test, where the t-statistics (p-values) are aggregated, for both the ordinary least squares and the corresponding White-adjusted estimates. The cumulative evidence rejects the null hypothesis of $\alpha_2 = 0$ at the 0,001 significance level in both instances.

Banker et al. (1994) also screen the data for outliers. They delete those who met all four of the Belsey-Kuh-Welsch criteria and rerun the regression [Belsey-Kuh-Welsch 1980, reported in Banker et al. (1994)]. The linearity assumption was rejected at the 5% level of significance in seven of the eleven datasets.

In addition, Data Envelopment Analysis (DEA) methodology was used to examine the hypothesis regarding linearity. The methodology *employs a non-parametric specification to estimate the production function (the function relating inputs to outputs) from observed data.* Estimates and test results are thus likely to be more robust (Banker et al. 1994: 279). The results of Banker et al. (1994) support a non-linear relationship between project size and effort.

The results published by Kitchenham (1992) and Banker et al. (1994) are examined in the next section.

5.4.3 REGRESSION ANALYSIS

5.4.3.1 Introduction

The regression analyses done by Kitchenham (1992) and Banker et al. (1994) have used *software size* as independent variable and *effort to develop the software* as the dependent variable.

Software size is usually measured as either lines of code or function points. The lines of code measure as an indication of size has been criticized in the literature (Jones 1986; Matson et al. 1994; Wellman 1993). The most commonly known cost models, e.g. COCOMO, use lines of code as indicator of size in their analysis. Seven of the eleven datasets used KLOC (thousands of lines of code) as software size measure. Function points are used in the remaining four datasets.

The inattention to the assumptions applicable to regression modelling and the number of observations on which the software cost estimation models are based are issues that have been raised by Matson et al. (1994). They discuss these issues for a function point dataset.

In order to establish empirically, for the given datasets, the aptness of using the regression model as an instrument to prove the linearity/nonlinearity of the software production function, the following aspects, for the linear and quadratic fit, are investigated:

1) The role of influential points
2) The $\mathbf{R}^2$ and **mse** measures
3) Residuals
4) Appropriateness and validity of t-tests.

5.4.3.2 Datasets used

The same datasets, with the exception of the MERMAID-3 and the Behrens dataset, that are used by Kitchenham (1992) and Banker et al. (1994), are used in the analysis. In addition, a Finnish dataset comprising 40 observations, was received from Dr. B. Kitchenham[2].

DATASET	DATE PUBLISHED	NUMBER OF OBSERVATIONS	TYPE OF SIZE MEASURE
Belady-Lehman	1979	33	Lines of code
Boehm (COCOMO)	1981	63	Lines of code
Yourdon	1981	17	Lines of code
Bailey-Basili	1981	19	Lines of code
Wingfield	1982	15	Lines of code
Albrecht-Gaffney	1983	24	Function points
Kitchenham-Taylor	1985	33	Lines of code
Kemerer	1987	15	Lines of code
MERMAID-1	1992	81	Function points
MERMAID-2	1992	30	Function points
FINN	1993	40	Function points

Table 5.4 The datasets

Two datasets (Wingfield and Yourdon) refer to projects developed in COBOL and are business applications. The Bailey-Basili dataset refers to projects developed in Fortran and is of a scientific nature. Boehm's dataset contains both COBOL (5 projects) and FORTRAN (24 projects) as well as other programming language projects. The Belady-Lehman dataset does not give the precise definitions used and the type of development is uncertain (Conte et al. 1986). Boehm's and Wingfield's datasets exclude comment lines in their lines of code count while the Bailey-Basili and Yourdon's datasets include comment lines. The Kitchenham dataset refers to 10 projects with S3 as programming language and the remaining 23 projects were developed mainly in COBOL with some in Assembler. Kemerer's dataset consists of 15 data processing development projects of which 12 are entirely written in COBOL (Kemerer 1987).

[2] The Finnish Dataset was provided by Sakari Kalliomaki, Hannu Maki and Kari Kansala to the MERMAID project.

Four datasets use function points as the size measure.

5.4.3.3 Scatterplots of the data

The scatterplots (EFFORT versus SIZE) indicate, in all instances, a clustering near the origin for small to medium size projects and a few isolated points for large projects. The scatterplots are in appendix D.

5.4.3.4 Results of linear regression analysis

The application of the logarithmic transformation changes the multiplicative model into an additive one. Such a nonlinear model is called intrinsically linear (Draper & Smith 1966: 132).

Table 5.5 lists the results.

DATA	VALUE OF b	STANDARD ERROR (s_b)	R^2 value for fitted model (%)	standard error of estimate
BELADY-LEHMAN	1.061	0.101	78.12	0.767
BOEHM	1.108	0.085	73.72	0.943
YOURDON	0.716	0.230	39.30	0.735
BAILEY-BASILI	0.951	0.068	91.93	0.331
WINGFIELD	1.059	0.294	50.06	0.710
ALBRECHT-GAFFNEY	1.487	0.191	73.48	0.615
KITCHENHAM-TAYLOR	0.816	0.166	43.76	0.862
KEMERER	0.815	0.178	61.71	0.581
MERMAID-1	0.941	0.107	49.55	0.593
MERMAID-2	0.824	0.135	57.12	0.905
FINN	1.058	0.156	56.09	0.792

Table 5.5 Linear regression analysis results

Table 5.6 lists the results of testing the hypothesis: $b = 1$ (using the 5% level of significance).

DATA	T-VALUE	CRITICAL VALUE	REJECT/DO NOT REJECT HYPOTHESIS
BELADY-LEHMAN	0.604	2.036	DO NOT REJECT
BOEHM	1.271	1.999	DO NOT REJECT
YOURDON	-1.235	2.120	DO NOT REJECT
BAILEY-BASILI	-0.721	2.101	DO NOT REJECT
WINGFIELD	0.201	2.145	DO NOT REJECT
ALBRECHT-GAFFNEY	2.55	2.069	DO REJECT
KITCHENHAM-TAYLOR	-1.108	2.036	DO NOT REJECT
KEMERER	-1,039	2.145	DO NOT REJECT
MERMAID-1	-0.551	1.993	DO NOT REJECT
MERMAID-2	-1.304	2.045	DO NOT REJECT
FINN	0.372	2.023	DO NOT REJECT

Table 5.6 Results of testing the hypothesis: $b = 1$

5.4.3.5 Results for the quadratic model

The following table list the results of fitting the model:

$$y = \alpha_0 + \alpha_1 x + \alpha_2 x^2$$

DATASET	α_0 ($s\alpha_0$) (t-value)	α_1 ($s\alpha_1$) (t-value)	α_2 ($s\alpha_2$) (t-value)	R^2 (ADJ) %	STANDARD ERROR OF ESTIMATE
BELADY-LEHMAN	-309.057 (369.821) (-0.8357)	17.571 (5.549) (3.1668)	-0.016 (0.009) (-1.751)	42.54	1564.397
BOEHM	-207.900 (190.252) (-1.093)	16.949 (2.699) (6.279)	-0.010 (0.003) (-3.123)	56.16	1206.108
YOURDON	1.506 (20.95) (0.072)	1.63 (0.937) (1.739)	-0.006 (0.007) (-0.808)	34.68	35.799
BAILEY-BASILI	-10.803 (7.951) (-1.359)	3.233 (0.533) (6.068)	-0.019 (0.006) (-3.07)	87.15	17.706
KITCHENHAM - TAYLOR	-33.336 (22.463) (-1.484)	12.396 3.354 (3.696)	0.271 0.088 (-3.073)	30.77	57.234
WINGFIELD	510.166 (818.736) (0.623)	0.083 (8.553) (0.01)	0.023 (0.018) (1.307)	66.68	897.692

ALBRECHT-GAFFNEY	8.478
	(4.535)
	(1.869)
		-0.014
		(0.013)
		(-1.093)
			0(3.4E-05)
			0(6.15E-06)
			(5.593)
				94.47	6.684

KEMERER	102.774 (118.041) (0.871)	-0,533 (1.227) (-0.434)	0.004 (0.003) (1.633)	54.37	177.702
MERMAID-1	521.646 (967.584) (0.539)	13.974 (5.144) (2.717)	0.004 (0.006) (0.745)	53.65	3008.255
MERMAID-2	-405.292 (2404.301) (-0.169)	44.947 (11.949) (3.761)	-0.025 (0.009) (-2.739)	35.27	8244.355
FINN	-3093.102 (2488.931) (-1.243)	22.365 (6.838) (3.271)	-0.008 (0.004) (-2.021)	44.50	5315.646

Table 5.7 Results of fitting the quadratic model

The standard error for each estimated parameter and the t-statistic are presented respectively in parentheses.

5.4.3.6 Discussion and conclusion

1 Influential points

The scatterplots of all the datasets exhibit the pattern of a cluster of projects of similar size and effort and a few isolated points. This is an inherent characteristic of software projects at a development organisation. Organisations have a bulk of projects that are small-to-medium projects with only a few (say three or four) very large projects. These few large project datapoints have a definite influence on the interpretation of the regression equation.

A datapoint is called influential if its removal from the dataset will substantially alter the results obtained for the full set of datapoints. The existence of these "influential points" will impact the regression equation in the following way: Extreme cases lead to an increase in the total variability. Geometrically, the cluster of points near the origin tend to be clumped together and "behave as a single point" (Matson et al. 1994).

Deleting the influential points will improve the fit of the model. However, it is of utmost importance to consider these points as they are an inherent part of project data from companies. To establish the true relationship between software size and software effort, they need to be taken into account.

2 R^2 and mean square error (mse)

The R^2 value, known as the coefficient of multiple determination, is usually used to determine the amount of variability in the dependent variable explained by the independent variable(s), giving an idea of the adequacy of the model.

Four datasets have a relatively high R^2 value (> 70%) when the multiplicative model was fitted and 3 datasets when the quadratic model was fitted to the data.

However, if regression theory assumptions are seriously violated, the R^2 value is of little importance. Furthermore, the existence of "extreme or influential points" greatly influenced the R^2 value. It is thus necessary to examine the residuals to determine whether the assumptions are violated.

The **mse** (mean square error) value is important from an estimation perspective. A smaller **mse** will result in narrower prediction intervals over the relevant range of the independent variable as the width of the prediction interval is primarily determined by the **mse** value. The square root of the **mse** is the standard error of the estimate, the value usually given in the output of statistical packages.

The standard errors of the estimate seem to be reasonably small for the fit of the multiplicative model in all 11 cases (remember that the values in the table are ln-values and need to be transformed back).

The standard error of the estimate seems to be higher in the case of the fit of the quadratic model.

3 Residuals

A graphical examination of residuals (see Appendix E) reveals the following:

a) **For the linear regression fit:**

The residual plots (plot of residuals versus the log of the independent variable), appear to fall in a horizontal band, except in the case of the Kemerer dataset, which exhibits a pattern indicating that the variance of the residuals is not constant.

From the normal probability plots it seems that normality can be assumed for the Basili, Boehm and Mermaid-2 datasets. For the remaining eight datasets normality cannot be reasonably assumed.

b) **For the quadratic fit:**

Residual plots of the residuals versus the predicted values indicate the presence of possible non-constant variance of the residuals. In addition, the normal probability plots indicate deviation from normality.

4 Appropriateness and validity of the t-tests.

Inferences concerning parameters are inaccurate if the model is misspecified (Banker et al. 1994) or if there is nonconformity of the residuals to the model assumptions (Matson et al. 1994). Furthermore, a large **mse**, along with serious violations of assumptions, renders the resulting inferences virtually meaningless.

Banker et al. (1994) base their use of the quadratic model rather than the transformed linear model on the ***assumption*** that the linear model is misspecified.

The regression analysis confirms that the hypothesis: $b = 1$, cannot be rejected in ten of the eleven datasets. However, as previously mentioned, violation regarding the distribution of the residuals, homoscedasticity and the existence of influential points lead one to query the validity of this method to establish whether a linear model is adequate.

CONCLUSIVE REMARKS

The lack of published data in this area is a well known fact. Research of this kind can only be extended once bigger and more recent datasets are available.

Of great concern is the age of the datasets. Software development technologies have changed dramatically over the last 10 years. The question regarding the relevance of the first 6 datasets is therefore pressing. Another important aspect raised by Conte et al. (1986) is the inclusion/exclusion of comments as lines of code. However, in order to compare the results published by Kitchenham (1992) and Banker et al. (1994) these datasets were analysed.

Regarding the results, it seems that neither the fit of the multiplicative model nor the quadratic model, using regression analysis, provide a satisfactory answer. The need for further research thus become apparent.

5.4.4 RESULTS OF THE ANALYSIS OF COMBINED DATASETS

The eleven datasets were combined into two datasets. The size measure for one dataset is function points and lines of code (in thousands) for the other dataset.

Dataset 1 will be called the lines of code dataset and dataset 2 the function point dataset.

5.4.4.1 Analysis of Dataset 1

The combined dataset consists of 195 datapoints. A scatterplot of development effort vs. lines of code is depicted in figure 5.6.

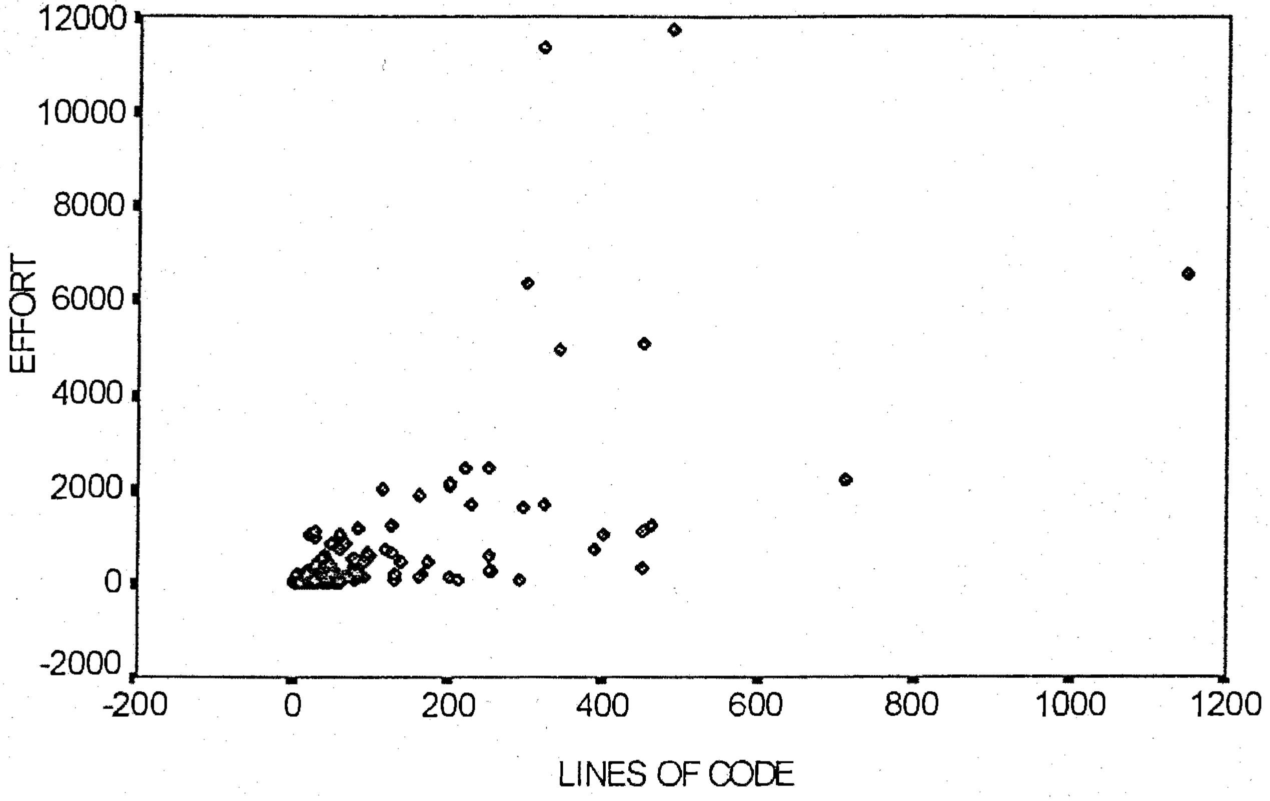


Figure 5.6 Scatterplot of Lines of Code vs. Development Effort

After examining the scatterplot it was decided that the data should be partitioned. A cutoff point of 250 lines of code was chosen as a different model seems to be appropriate for the fifteen datapoints above 250. This results in a dataset of 180 points which covers 92% of the original dataset. In the discussion that follows, the reduced dataset of 180 points will be used.

A kernel-type scatterplot smoother, an explanatory data-analytical tool, was used to examine the data in order to determine if the assumptions normally applicable to linear and quadratic regression are reasonably met.

Smoothing is an approach that relies on the data to specify the form of the model. It *fits a curve to the data locally, so that at any point the curve at that point depends only on the observations at that point and some specified neighbouring points* (S/PLUS for Windows ... 1994: 9-44). The estimate of the response is thus less variable than the original observed response, hence the name "smoother". The procedures for producing these fits are called scatterplot smoothers, with the kernel-type being one such smoother.

The kernel-type smoother is a type of local average that, for each target point x_i in the predictor space, calculates a weighted average $\hat{y}_i$ of the observations in a neighbourhood of the target point:

$$\hat{y}_i = \sum_{j=1}^{n} w_{ij} y_j \qquad i = 1, 2, \ldots, n$$

$$where \quad w_{ij} = \frac{K(\frac{x_i - x_j}{b})}{\sum_{j=1}^{n} K(\frac{x_i - x_j}{b})}$$

are weights which sums to one:

$$\sum_{j=1}^{n} w_{ij} = 1$$

The function K that is used to calculate the weights is called a kernel function, which typically has the following properties:

(a) $K(t) > 0$ for all t

(b) $\int_{-\infty}^{\infty} K(t)dt = 1$

(c) $K(-t) = K(t)$ *for all* t (symmetry).

The parameter b is the bandwidth parameter, which determines how large a neighbourhood of the target point is used to calculate the local average. Large bandwidths generate a smoother curve.

With a kernel estimate, the values of y_j for which the x_j*'s* are close to x_i, , get relatively larger weights, while values of y_j for which the x_j's are far from x_i get small or zero weights. The bandwidth parameter b determines the width of $K(t/b)$, and hence controls the size of the region around x_i for which y_j receives relatively large weights.

The "normal" kernel was chosen where

$$K_{nor}(t) = \frac{1}{\sqrt{2\pi}(0.37)} \exp\left[\frac{-t^2}{2(0.37)^2}\right]$$

The bandwidth was chosen as 100.

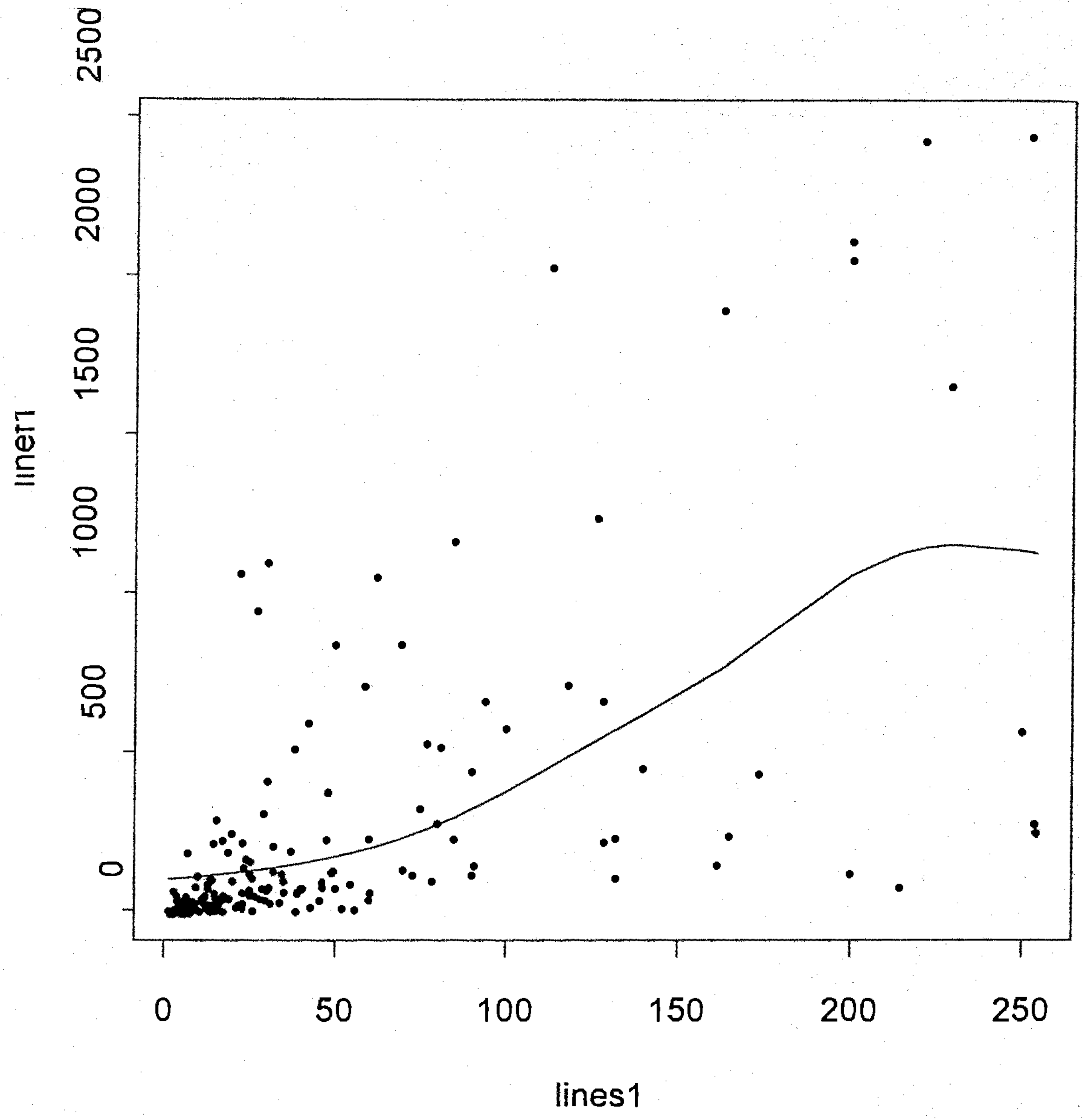


Figure 5.7 Kernel Scatterplot smoother applied to Lines of Code dataset

Kernel = Normal: Bandwidth = 100

The plot confirms the violation of homoscedasticity, i.e. we cannot assume constant variance. The variance increases proportionally to the mean.

THE QUASI-POISSON REGRESSION MODEL

The generalized linear model provides a way to estimate a function (called the link function) of the mean response as a linear function of the values of a set of p predictors. It is written as:

$$g(E(Y|x)) = g(\mu) = \beta_0 + \sum_{i=1}^{P} \beta_i x_i = \eta(x)$$

where g is the link function. The linear function of the predictors, $\eta(x)$, is called the linear predictor. For the generalized linear model, the variance of Y may be a function of the mean response μ.

$$var\ (Y) = \phi\, V(\mu).$$

Poisson regression is a special case of the generalized linear model.

For the Poisson regression model we have that

$$g(\mu) = log(\mu)$$

and the variance is defined by

$$var(Y) = \phi\mu.$$

We have decided to use Poisson regression for the following reasons:

a) The response variable (development effort = number of man-months) is a “count” type of response.

b) The scatterplot has revealed that the variance increases proportionally with the mean.

c) The scatterplot smoother suggested that the mean was not linear in our independent variable, i.e. the number of lines of code (in thousands).

As the data was over dispersed, we cannot assume that $\phi = 1$. We use quasi-likelihood estimation as it allows us to estimate the dispersion in under- or over-dispersed regression models.

For our dataset we have that $var(Y|x) = 364.64$ (the estimated over dispersion).

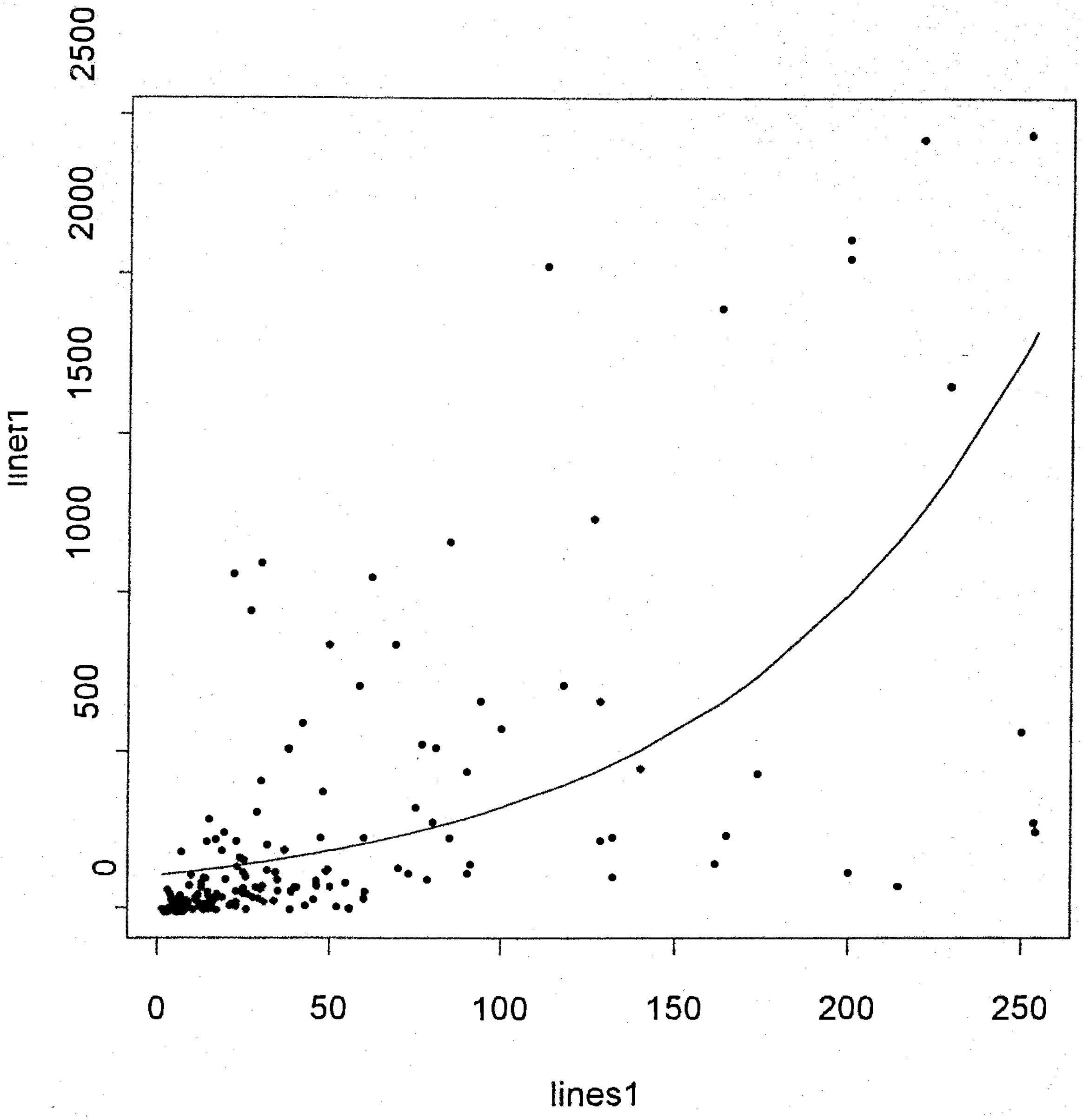


Figure 5.8 Quasi-Poisson Model fitted to Lines of Code dataset

An assessment of the model can be made as follows:

SOURCE	DEVIANCE	df
Model (Difference in deviance)	38722.19	1
Residual (Fitted deviance)	48659.81	178
Total (Null deviance)	87382.00	179

Table 5.8 Assessment of the model applied to dataset 1

From the above we can test the hypothesis H_0: the model has no contribution.

The difference in deviance (due to the model) is asymptotically chi-square distributed with 1 degree of freedom. We can see that the value is highly significant, thus implying that the model contributes to explain the relationship.

5.4.4.2 Analysis of Dataset 2

The combined dataset consists of 188 datapoints. A scatterplot of development effort (in hours) vs. function points is given in figure 5.9.

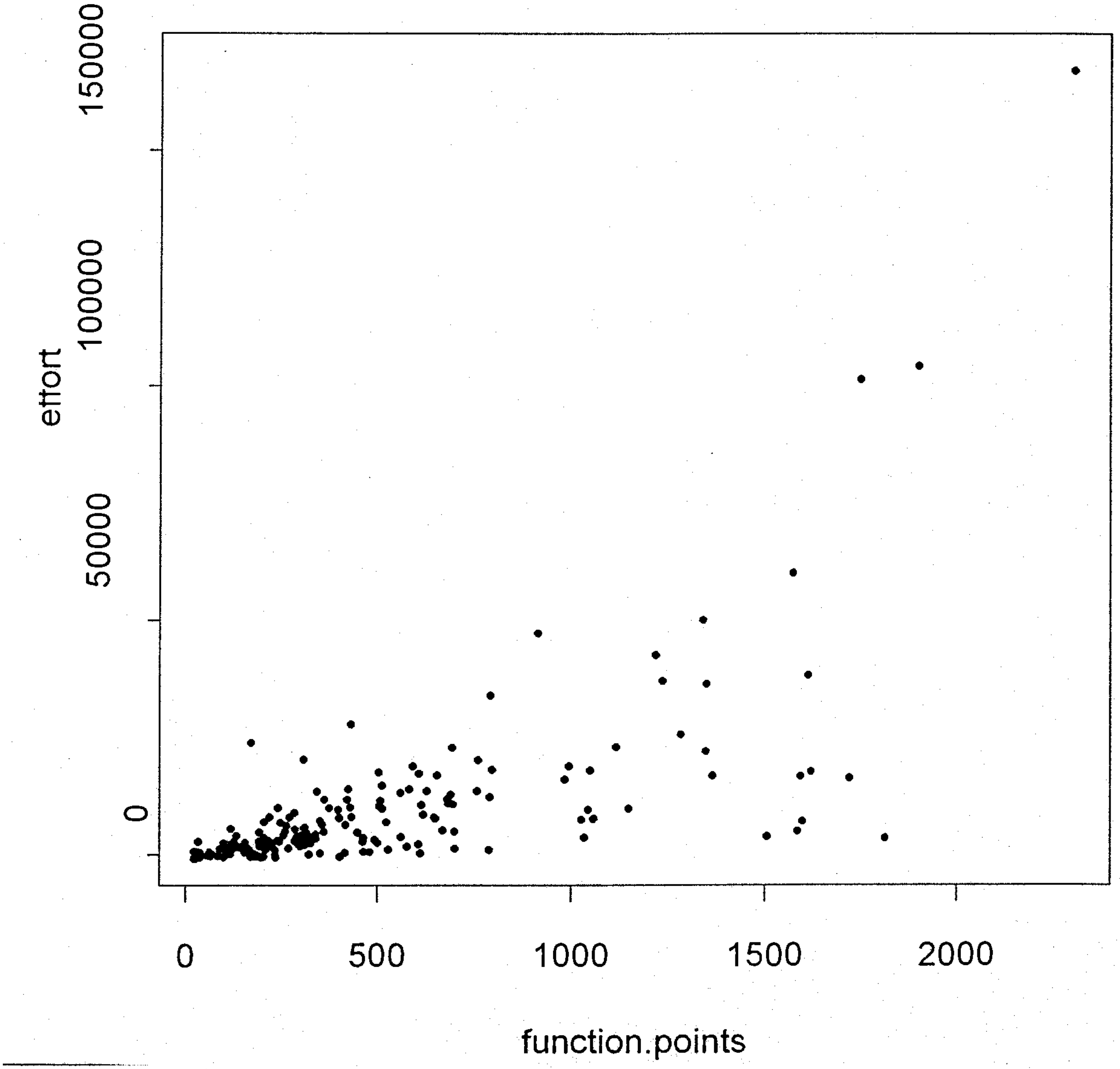


Figure 5.9 Scatterplot of function points vs. development effort

A "normal" kernel-type scatterplot smoother was applied with a bandwidth of 500.

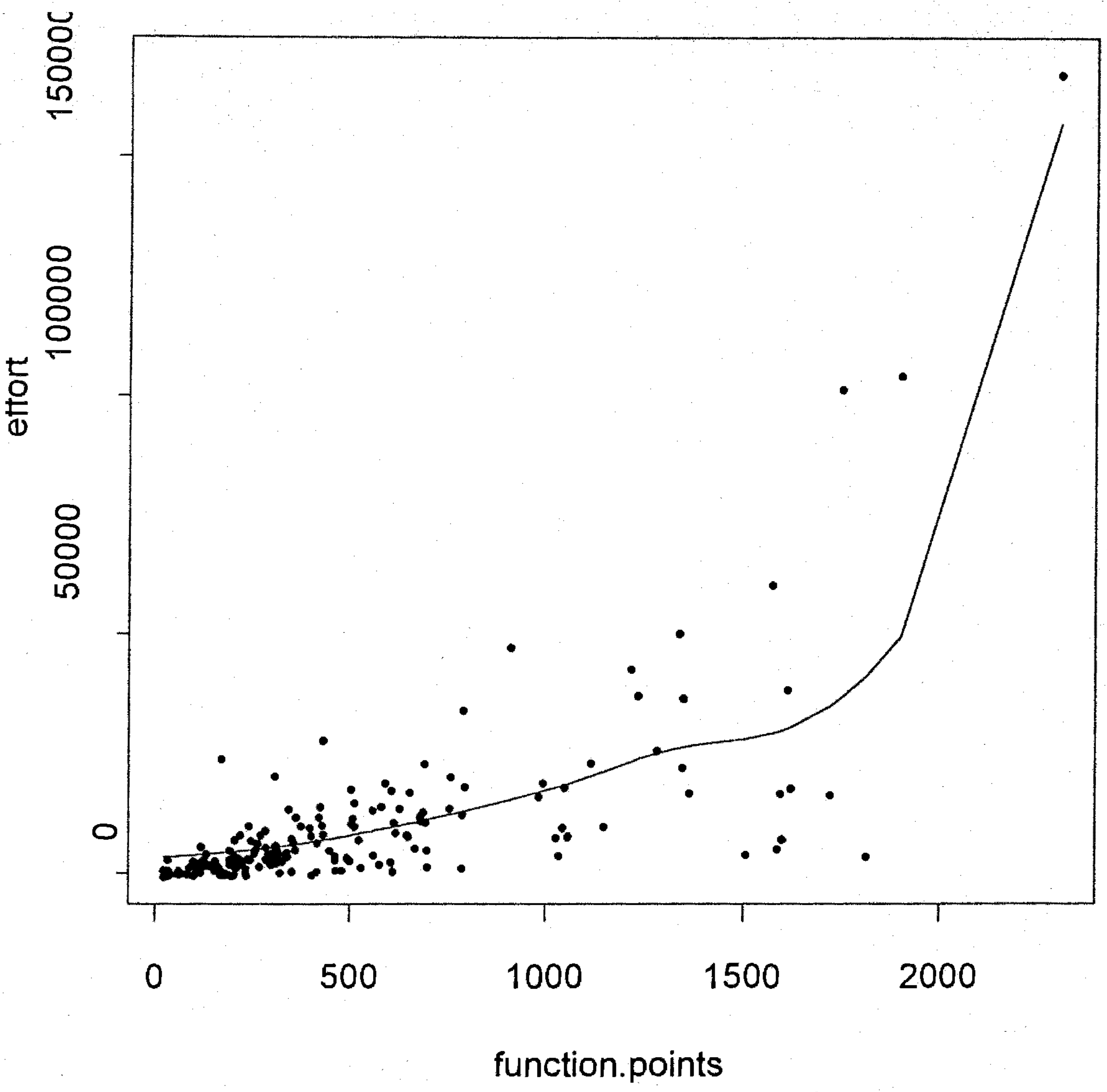


Figure 5.10 Kernel scatterplot smoother applied to function points dataset

The plot once again confirms the violation of homoscedasticity, i.e. we cannot assume constant variance. The variance increases proportionally to the mean.

Poisson regression is once again used as the same reasons that applied to the lines of code dataset holds true for the function point dataset.

As the data was over dispersed, we cannot assume that $\phi = 1$.

For our dataset we have that $var(Y|x) = 5704.557$ (the estimated over dispersion).

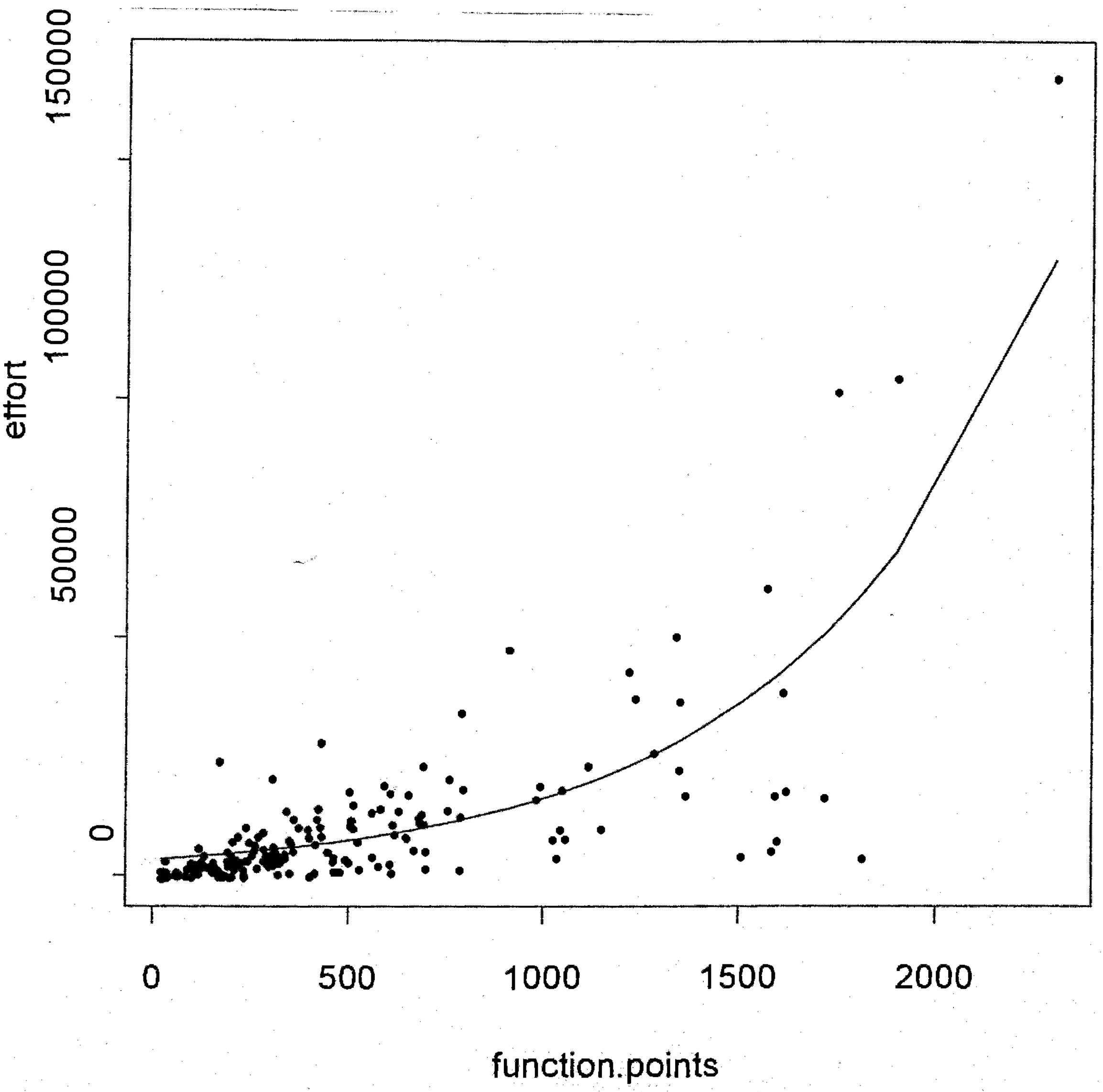


Figure 5.11 Quasi-Poisson model fitted to function points dataset

An assessment of the model is:

SOURCE	DEVIANCE	df
Model (Difference in deviance)	1847144.00	1
Residual (Fitted deviance)	1013211.00	186
Total (Null deviance)	2860355.00	187

Table 5.9 Assessment of the model applied to dataset 2

The value is highly significant, thus implying that the model contributes to explain the relationship.

5.4.4.3 Summary and conclusion

The eleven individual datasets were combined according to the size measure. The scatterplots (Figure 5.6 and Figure 5.9) of the combined datasets confirm that, in both cases, the same type of relationship holds true as that which applies for the individual datasets. One can thus use the combined datasets to model the relationship. An advantage of combining the datasets is the bigger sample size.

Through the application of a scatterplot smoother it was clear that, in both datasets, the variance increases proportionally to the mean. Furthermore, as we are working with "count" type data, it was decided to apply Poisson regression modelling to the data. As the data was over dispersed, quasi-likelihood estimation was used.

The models fitted seem to fit the data reasonably well. However, further research is needed for the refinement and to establish the overall validity of the models. Books that can be consulted in this regard are Chambers & Hastie (1992) and Venables & Ripley (1994).

5.4.5 DIRECTIONS FOR FURTHER RESEARCH

The following represents brief descriptions of important avenues for further investigation that have opened due to recent developments/research:

1) Capacci and Stamelos (1993) examine the use of artificial neural networks and factor analysis in the construction of software cost estimation models. They describe artificial neural networks as *structures with many degrees of freedom which, by calibration of a set of parameters, can fit almost all kinds of analytic functions.* Their results, based on an experimental dataset, showed that the neural network model performs better than the models used (Capacci & Stamelos 1993). They propose further research into the applicability of these two techniques in software cost estimation. Their work is important to the linear/nonlinear debate as it suggests

   - that we need to look beyond one input variable (size) to determine the output (effort) variable and
   - it is the first application of neural networks to the software cost estimation modelling problem.

2) Matson et al. (1994) recommend the *unbundling of the function point variable into its constituent components* to achieve more precise prediction of the effort needed, supporting the view of Capacci and Stamelos (1993). Matson et al. (1994) believe that better use can be made of available information to markedly improve cost estimation modelling.

3) Miyazaki et al. (1991) have suggested the use of the least squares method (which minimizes the sum of squares of R_i (the relative error) instead of the ordinary least squares method (which minimizes the sum of squares of errors), to estimate the parameter values in a software prediction model. The relative error is defined as

$$RY_i = (Z_i - Y_i) / Y_i$$

where Z_i is the estimated value of a dependent variable and Y_i is the actual value of the variable for the i'th sample datapoint.

4) Abdel-Hamid (1990) has suggested a system-dynamic simulation approach to software project cost estimation. He argues that raw historical project results do not necessarily constitute the best data to be used for future estimation.

5.5 SOFTWARE COST ESTIMATION MODELS AND PROJECT MANAGEMENT TECHNIQUES - FRIENDS OR FOES?

5.5.1 INTRODUCTION

Software cost estimation models and project management techniques as applied to software are both well researched topics. What is lacking however, is the relationship/connection between software cost estimation models and project management techniques such as PERT (Program Evaluation and Review Technique). Wellman (1993) remarks that cost and resource estimates are prepared separately from project schedules which leads to inconsistency. As overruns in time and budget on software development projects keep on occurring, a better understanding of this relationship is urgently required.

In section 5.5.2 a short summary is provided regarding aspects of project management techniques pertaining to software development. Software cost estimation models were discussed in 5.3.5. Section 5.5.3 discusses and explores the relationship and suggests several aspects that require further investigation. Finally, in section 5.5.4 a comparison is made between software cost estimation models and project management techniques.

5.5.2 PROJECT MANAGEMENT TECHNIQUES

Project management techniques are employed to aid project managers in the planning, scheduling and control aspects of a project.

Project management is defined by Edwards and Moores (1994: 139) as ***the deployment of project resources between start and end points of a project in such a way that a specified goal is achieved***

Lee and Murata (1994: 150) define **software project management** as a ***system of procedures, practices, technologies, and know-how that provide the planning, organisation, staffing, direction, and control necessary to successfully manage a software development project with***

given resources.

Important preconditions for effective project management include the following:

1) that the project goal, the start and the end points are specified clearly at the beginning;
2) that the resources allocated are sufficient and
3) that neither of the previous two aspects change significantly during the life-time of the project (Edwards & Moores 1994).

The initial requirements from the software user usually change during software development. Change during software development is therefore the rule rather than the exception. In addition, there is usually tension between demands for *higher quality, more functionality, reduced development time and lower costs* (Edwards & Moores 1994: 140). Software development projects thus constitutes more uncertainties to the project manager than most other project types.

Project management techniques that are used in software development organisations include the work breakdown structure (WBS), Gannt charts and project network diagrams such as PERT (Kidd 1991). In a survey conducted by Moores and Edwards (1992) they found that planning is done exclusively with software project management tools. This indicates the extent of the impact of these tools in the industry.

Project management techniques such as PERT, CPM and Gannt charts concentrate on the scheduling of activities (Lee et al. 1994). Lee et al. (1994) argue that the techniques and models suggested for project management have the following shortcomings regarding software development:

1) they do not provide the information needed by the manager to analyse the progress of activities
2) they cannot represent the hierarchical relationship of activities and subactivities as an integral system component
3) activity dependencies do not include the notion of boolean conditions
4) they cannot represent the rescheduled activity when a completed activity is being

reactivated

5) they cannot provide the manager with information when an activity is activated before all prior activities have been completed

6) they are inadequate for representing the criteria that trigger the start of an activity.

5.5.3 SOFTWARE COST ESTIMATION MODELS AND PROJECT MANAGEMENT TECHNIQUES - ARE THERE SYNERGY?

Software cost estimation models and tools were generally developed for estimation at the macro level and are not specifically task-oriented. The detailed version of COCOMO attempts aspects of task-orientation by introducing phase-sensitive effort multipliers (a set of phase-sensitive effort multipliers is available for each cost driver in the model) and a three-level product hierarchy (module, subsystem and system levels of the product are acknowledged and ratings of the cost driver can be made at the appropriate level). It can thus be seen as a micro model, i.e. one that uses the bottom-up approach to estimation.

However, very few project managers use cost estimating tools in industry (Lederer & Prasad 1992; Van Genuchten & Koolen 1991) as opposed to project management tools (Moores & Edwards 1992).

Criticisms against software cost estimation models (Edwards & Moores 1994) include:

1) different models provide "very different" cost estimates for the same data

2) calibrating the model to the specific environment in which it is to be used is essential but it is not always clear how the model relates to other environments so that constructive calibration can take place

3) the use of adjustment factors is subjective and will vary between users.

Edwards and Moores (1994) discuss the conflict between the use of estimating and planning tools in software development management. Project management tools are specifically task-oriented. In the use of these tools, it is necessary to estimate the time, cost and performance for each identified task. Edwards and Moores (1994) stress that project management techniques do not support the ***determination*** of the relevant estimates necessary to apply the technique and argue that estimation tools do play a useful role and is required in project management. Carter et al. (1987), when discussing the estimation of time scales, also emphasize this aspect, mentioning that accurate estimation of task time schedules for software development projects is "fraught with difficulties". They conclude: *the accurate estimation of time scales and costs creates fundamental problems that require considerable research before the network analysis techniques can realise their full potential in the successful control of computer projects* (Carter, Clare & Thorogood 1987: 150). Wellman (1993) remarks that estimating is not yet established as a skill base within software engineering. Accuracy of estimation will ultimately influence the successfulness of the PERT or other project management techniques.

Edwards and Moores (1994) propose a EEPS (early estimating and planning stages) model. This model focuses on the involvement of the client in the negotiation of the cost and functionality of a proposed system and is described in Edwards and Moores (1994).

Figure 5.12 depicts a diagrammatic description of the model.

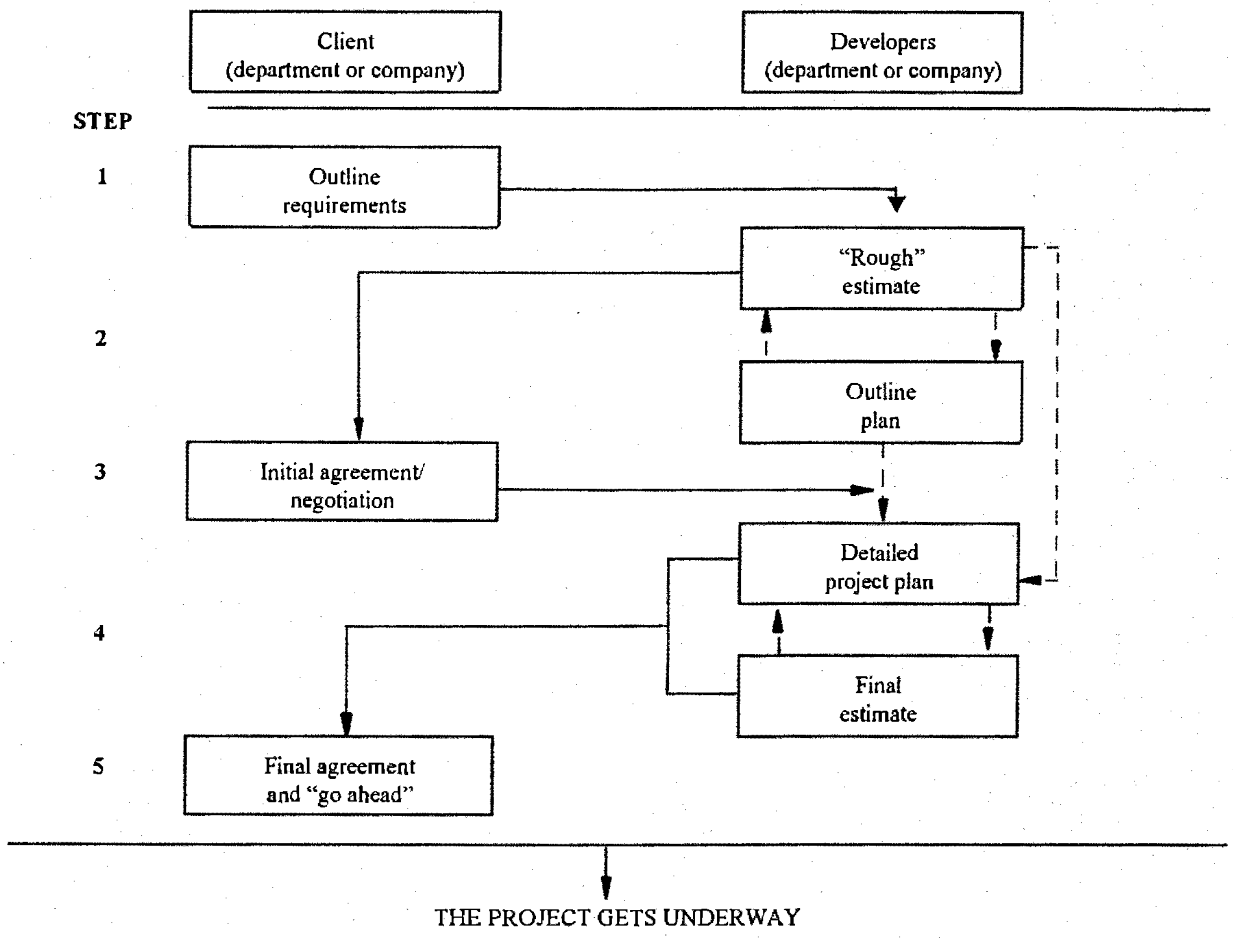


Figure 5.12 The early estimating and planning stages (EEPS) model

In their model, Edwards and Moores (1994: 142) distinguish between what they call a "rough" estimate (made at the beginning of the project when only vague requirements are available) and an estimate made when detailed plans and requirement documents are available. These estimates differ with respect to:

1) *the level of information available*

2) *the people involved in the discussion regarding the estimate and*

3) *the use to which the information is put.*

Most current software cost estimating tools are useful if one looks at estimation from the first perspective (i.e. a "rough guide"). However, if estimation is seen as a "bottom-up, plan-based number-assigning task" (Edwards & Moores 1994) the following aspects are emphasized:

1) current software cost estimation models are addressing the problem from top-down instead of bottom-up;
2) the initial estimate is not perceived as a true estimate and
3) estimation is seen to take place at the planning stage where project management tools are used and not software cost estimation tools.

Edwards and Moores (1994) conducted a telephone survey regarding the estimation/planning conflict on 17 companies. They conclude that there is contradiction in project managers' minds regarding the concept of an estimate. They do make an early "estimate", but see estimation mostly as a bottom-up task, thus implying that it is subsumed within the planning process. Project management tools are thus used as estimating support tools. As estimation is still perceived to be the problem by most managers (Moores & Edwards 1992), they prefer to use planning tools which do not constrain the way in which an estimate is produced.

Edwards and Moores (1994) emphasize that the difference between planning and estimation tools lies in the way that they are used to model the cost of the project. According to Edwards and Moores (1994: 144), a planning tool provides a framework to model the project in terms of tasks that are based "in an unspecified manner on historical data" while an estimation tool imposes a model of the development process and requires information pertaining to the specific project under question.

Edwards and Moores (1994) conclude that existing estimating tools were not developed to address the need of the project manager and that there is a need to redefine the role and function of estimation models (and tools) to determine the place of these in project management.

They suggest the development of task-based estimating tools. The advantages of such an estimating tool will be:

1) that it provide a sound basis for incorporating information such as system size and productivity rate information,

2) that it provide estimates of some of the important product measures as the estimation is based on historical data and

3) it make use of the data available at the early requirement stage.

No evidence was found of data interchange between software cost estimation models and project planning systems (Wellman 1993). He adds that such an interface should not pose any difficulties. Westney (1989: 28-29) suggests seven possible approaches to the integration of estimating and planning. They are:

1) Use design information to generate both planning and estimating data simultaneously.

2) Provide an estimating database applications-writing capability.

3) Use the planning software as a method for estimating.

4) Export the information from planning software to general-purpose software.

5) Integrate estimating data with a scheduling algorithm in the same program.

6) Provide a flexible user-defined estimating program with export capabilities to planning and scheduling software.

7) Provide an interface between estimating and planning software.

It is interesting to note that Wellman (1993) when citing the seven approaches for linking estimating and planning tools by Westney (1989), comments that the third approach, namely to use the planning software as a method for estimating, is unsatisfactory for software development. This is exactly the same conclusion reached by Carter et al. (1987) and Edwards & Moores (1994) regarding the value of estimates from planning tools mentioned previously.

The approach to provide a flexible user-defined estimating program with export capabilities to planning and scheduling software, can possibly be achieved by using the MERMAID MARK 1P tool which was discussed in 5.3.6.

Wellman (1993: 64) presents an illustration where design, estimating and planning are carried out as complimentary activities. He stresses that modularity and constructability of a system should be compatible with the input to estimating and planning and vice versa, to ensure that iteration can be carried out easily and consistently.

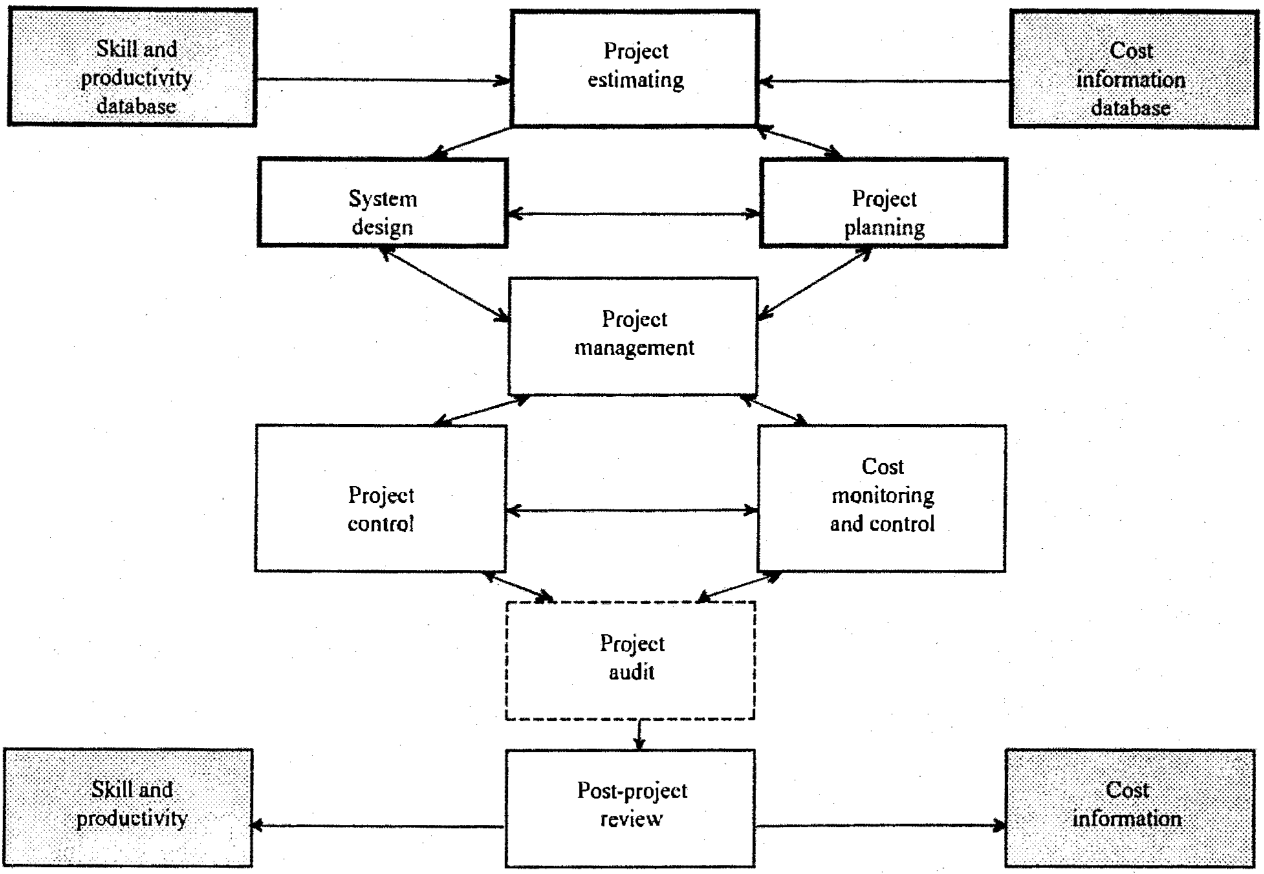


Figure 5.13 Estimator's view of a project

Several distinct research possibilities thus exist to study the synergy:

1) to develop task-based estimating tools as proposed by Edwards and Moores 1994.
2) to create a structure within which both types of tools exist and interface. The idea of using a cascade of techniques has been mentioned by Ariflugo (1993) and Heemstra (1992) with regard to the use of estimating tools. The structure could be extended to incorporate project management tools.
3) to develop a project management model that includes cost and duration aspects as well as complex relationships between activities. One such attempt is the Beta-distributed stochastic Petri-net model for software project time/cost management (Lee & Murata 1994). This model, as well as the PM-Net model (Lee et al. 1994), developed for software

project control, are briefly described in chapter 6.

4) to develop a concurrent project management model that deals with multiple projects. One such attempt is the work by Abdel-Hamid (1993) in his article: "A multiproject perspective of single-project dynamics".

5) to develop a combined model for software cost estimation and project management applicable to an object-oriented software development project

6) to investigate the quantification aspect of uncertainty in activity durations

7) to study the effect of change in requirements on software cost estimation models and project management techniques by means of scientific experimentation

Only the aspect of the quantification of uncertainty in activity durations will be investigated in chapter 6. However, task-based estimating and project management techniques are interrelated to this aspect and will be dealt with within the framework of addressing the uncertainty aspect.

5.5.4 COMPARISON BETWEEN ESTIMATING AND PROJECT MANAGEMENT TOOLS

The following table compares characteristics of project management and estimation tools.

CHARACTERISTIC	PROJECT MANAGEMENT TOOLS	ESTIMATION TOOLS
Structure	task-based	model-based
Change in requirements	Will relate to changing schedules	Parameters of model need to be changed.
Coverage	focus on all aspects of project	have focused only on code aspects of the software development process
Focus on	actions to accomplish tasks	things ; resource by quantity
Data requirements	only use subjective estimates made by personnel involved	use historical data on previous project

Table 5.10 Comparison of characteristics of project management and estimation tools

6. QUANTIFICATION ASPECTS OF UNCERTAINTY IN ACTIVITY DURATIONS

"If you knew Time as well as I do," said the Hatter, "you wouldn't talk about wasting it. It's ***him.****"*

"I don't know what you mean," said Alice.

"Of course you don't," the Hatter said, tossing his head contemptuously.

"I dare say you never even spoke to Time!"

"Perhaps not," Alice cautiously replied, "but I know I have to beat time when I learn music."

"Ah! That accounts for it" said the Hatter. "He won't stand beating. Now, if you only keep on good terms with him, he'd do almost anything you liked with the clock. For instance, suppose it were nine o'clock in the morning, just time to begin lessons: you'd only have to whisper a hint at Time, and round goes the clock in a twinkling! Half-past one, time for dinner!"

("I only wish it was," the March Hare said to itself in a whisper.)

"That would be grand, certainly," said Alice thoughtfully; "but then -

I shouldn't be hungry for it, you know."

"Not at first, perhaps," said the Hatter: "but you could keep it to half-past one as long as you liked."

"Is that the way you manage?" Alice asked.

Alice in Wonderland (Lewis Carroll 1865)

6.1 INTRODUCTION

The management of time in organisations may often be done in the haphazard way described in the situation above.

Proper project planning is a key success factor in organisations. Poor project planning can have devastating effects. It is of strategic importance to have systems in place for good project planning. A very important aspect of project planning is scheduling. In scheduling, the quantification of uncertainty, regarding the duration of activities that constitute the project, as well as the project completion time, is of paramount importance. It is a key determinant of the success of scheduling techniques. As such, the quantification of uncertainty in activity durations is a strategic measurement issue that will ultimately influence the quality of the end product. Finley and Fisher (1994: 27) remarks: *Dealing with risk requires determining the amount of uncertainty. Simply saying that too much effort is needed to quantify the amount of uncertainty does not make the uncertainty to go away; this attitude probably contributes to the level of actual risk because management has no knowledge of what is lurking in the future.*

Uncertainty in activity durations as well as in resource availability and/or cash flows is one of the current issues of interest to managers cited in a review of recent developments in activity networks (Elmaghraby 1995).

The estimation of activity durations is a critical aspect of project planning, as these estimations are the basic input for project scheduling techniques. Carter et al. (1987), Edwards & Moores (1994) and Wellman (1993) all emphasized the need for more accurate estimation of activity time, and thus cost, specifically for software projects. Whitten (1995: 105) remarks: *Estimating the duration of an activity is perhaps the most difficult task in developing the project scheduling plan.*

The quantification of uncertainty in activity durations will be discussed as follows:

1) definitions of terms used in the chapter
2) activity duration estimates
3) a review and comparison of proposed activity duration distributions
4) selection criteria for activity duration distributions
5) a review on estimation issues relating to activity duration.

The use of simulation, particularly the application of Monte Carlo methods, in the analysis of stochastic networks for project scheduling, is an important field of research, but will not be discussed in this dissertation. Developments like GERT (Graphical Evaluation and Review Technique) (Pritsker & Happ (1966) and Pritsker & Whitehouse (1966)), which aims to analyze networks with stochastic and logical properties as well as VERT (Venture Evaluation and Review Technique), developed by Moeller (1972) [reported in Kidd 1987], and discussed in comparison to other methods by Kidd (1987), which aims to jointly deal with uncertainty in time, cost and performance, are taken note of, but will not be discussed.

Related areas, i.e. research regarding resource-optimization and trade-off models (nonlinear (Deckro et al. 1995) and linear) are not included in the study.

An extensive bibliography of research on stochastic PERT networks for the period 1966-1987 can be found in Adlakha and Kulkarni (1989).

6.2 DEFINITIONS

6.2.1 ACTIVITY

An **activity** is defined as ***any undertaking that consumes time and resources*** (Elmaghraby 1977: 1).

6.2.2 EVENT

An **event** is ***a well-defined occurrence in time*** (Elmaghraby 1977: 1).

6.2.3 PROJECT

A collection of activities and events (Elmaghraby 1977: 1).

6.2.4 AN ACTIVITY NETWORK

An **activity network** is obtained ***when all the activities and events in a program are linked***

together sequentially in a proper relationship (Archibald & Villoria 1967: 16).

Generally, the line or arrow in a network represents a time-consuming activity and the circle or rectangle represents the event or node marking the beginning or end of an activity (Archibald & Villoria 1967).

6.3 ACTIVITY DURATION ESTIMATES

The input data values for activity durations consist of judgmental estimates made by so-called experts and are, as such, subjective in nature. The PERT technique, e.g. requires an estimate of the most likely, optimistic and pessimistic time for every activity duration.

Laranjeira (1990) argues that current experience does not confirm the PERT assumption that estimates are unbiased toward underestimation or overestimation. In his example, 12 out of 16 projects' size were underestimated. He attributes this to aspects such as lack of experience and/or knowledge, incomplete recall of historical projects and the desire to please management. This aspect is also raised by Pollack-Johnson (1995), who concludes that individual judgment has a tendency to be biased (usually towards an optimistic estimate). He advises that it can be improved by tracking historical performance of projects and adjusting estimates as needed with the help of formal methods such as bootstrapping.

Another aspect cited by Laranjeira (1990) concerning subjective judgement is that there is a wide variation due to psychological and personal factors, and thus estimates with required accuracy cannot be obtained.

Merkhofer (1987) warns against the use of words to communicate uncertainty in decision analysis, e.g. "almost certain to occur" as different people assign very different probabilities to such a statement. He introduces the method of probability encoding to quantify judgemental uncertainty. It is based on a structured interview between a trained interviewer and the person who needs to make the judgement.

Keefer and Verdini (1993) note that judgmental estimates of the 0.01 and 0.99 fractiles are very difficult. Accuracy and reliability of such assessments are not as good as for points removed further from the extreme. They argue that judgmental estimates of the median would be preferable to the mode and that the 0.05 and 0.95 or the 0.10 and the 0.90 fractiles are preferable to the 0.01 and 0.99 fractiles.

The complexity and dynamics of a judgmental estimate can be understood through the words of Robb Ware (Ware 1994: 10): *There is a great deal of difference in estimating the duration of something you merely observe, and something over which you have control.* Ware argues that accurate estimates is a function of technology and experience while control over the process is a function of influencing, motivating and steering people.

The underlying subjectivity is thus an integral part of the estimation process and should be acknowledged.

6.4 A REVIEW OF ACTIVITY DURATION DISTRIBUTIONS

6.4.1 INTRODUCTION

If the duration of an activity is uncertain, it implies that the activity duration (or activity completion time) is a random variable. Project risk analysis requires an apriori statistical distribution for activity durations. In particular, it is concerned with "combinations of distribution 'tails'" thus the shape of the distribution is important (Williams 1992: 267)

It is thus necessary to investigate the statistical distributions for activity durations suggested in the literature. This is extremely important as the activity duration distribution also determines the distribution of project completion time, a strategic component of project planning.

In the case of activity durations, little formal sample information is available to "fit" the distribution to, and subjective knowledge of the process needs to be utilized. This is usually in the form of an expert's perception of the cumulative distribution function (Lau & Somarajan 1995).

The human judgment in estimation adds another dimension to uncertainty in activity networks and needs to be acknowledged when quantifying uncertainty.

Debate on the form of the distribution for activity durations have been going on since the time that the first article on PERT (Program and Evaluation Review Technique) was published by Malcolm et al. (1959). Elmaghraby's (1977: 230) remark: *In the absence of any empirical evidence on the most appropriate form of the DF* (distribution function) *of* Y_u (Y_u is used to denote the duration of an activity), *there seems to be no compelling reason to adopt the one proposed by the originators of PERT!* depicts the controversy.

The proposed distributions, from the Beta distribution suggested initially by the original authors of the PERT technique (Malcolm et al. 1959) to the most current suggested distributions, the Erlang distribution (Bendell, Solomon & Carter 1995) and the Ramberg-Schmeiser distribution (Lau & Somarajan 1995) are discussed. A recent article (Mon, Cheng & Lu 1995) describes the application of fuzzy distributions as activity duration distributions. The author takes note of this development, but will not pursue it in this dissertation.

6.4.2 THE BETA DISTRIBUTION AND PERT

One of the best known and most commonly used activity network techniques, PERT, employ three time estimates for the time of each activity in the network. They are:

1) the optimistic time estimate - *a*
2) the pessimistic time estimate - *b*
3) the most-likely time estimate - *m*

These times are estimated by the project manager or responsible person, and are as such, judgemental and subjective in nature.

In the classic PERT approach, it is assumed:

1) that activity completion time (activity duration) follows a Beta-distribution with

$$f(x) = K(x-a)^{\alpha} (b-x)^{\gamma} \qquad a \leq x \leq b$$
$$\alpha_1 \gamma > -1.$$

We thus have four parameters: *a, b, α* and *γ* and *K* is a normalizing constant. The above expression can be transformed to the standardized Beta distribution

$$f(x) = \frac{1}{\beta(\alpha+1, \gamma+1)} x^{\alpha}(1-x)^{\gamma} \qquad 0 < x < 1$$

2) that the mean and variance are estimated by using the approximations:

$$\mu = \frac{(a+4m+b)}{6}$$

$$\sigma^2 = (\frac{b-a}{6})^2$$

The original PERT model also assumes (Elmaghraby 1977):

1) The activities are independent.
2) The critical path contains a large number of activities thus the Central Limit Theorem can be applied (When the estimates of activity duration times along each possible path of the network is added together, the **critical path** will *be the path that will consume the most time in reaching the end event* (Archibald & Villoria 1967: 19).

The original article on PERT by Malcolm et al. (1959) gives no justification for using the Beta distribution, but only states that it is an adequate statistical distribution to represent activity time. A later article by Clark (1962), one of the original authors, points out that PERT requires the expected time and standard deviation of an activity, and that the beta distribution fit the way the estimates were made, i.e. the estimates were to be made periodically and at low cost, and it was suggested that the time estimate that comes first to a persons mind would be the most likely time and the extreme minimum and maximum could also be estimated with some degree of accuracy.

No reported figures of accuracy are, however, given. Clark (1962) admits that he had no information on activity durations distributions and did not imply that the beta was the appropriate distribution. Clark (1962) thus acknowledges the essentially unsupported nature of the assumptions that have become standard in PERT.

Sasieni (1986: 1652) cites some advantages of the Beta distribution, namely

1) *it has a very flexible form*
2) *it can be given an arbitrary finite range*
3) *its shape varies from J with the maximum at either end of the range to unimodal with the mode at neither extreme, or even bimodal.*

The PERT assumptions have been extensively studied and criticised in the literature. (Bonett & Deckro 1993; Donaldson 1965; Grubbs 1962; MacCrimmon & Ryavec 1964).

Elmaghraby (1977) shows that PERT restricts the shape of the probability distribution that represents the uncertainty in activity durations. The simplifying assumption in the approximation of the expected value restricts the shape to only one of three, namely those of skewness $\pm \frac{1}{\sqrt{2}}$ or 0 (Elmaghraby 1977; Ranasinghe 1994).

Another shortcoming of the Beta distribution is that it does not cover all the possible "shapes" ((β_1, β_2) values) that can be assumed. Lau and Somarajan (1995) proposed the use of the Ramberg-Schmeiser distribution as supplementary to the B-distribution to accommodate the (β_1, β_2) values that are not included in the B-distribution. The Ramberg-Schmeiser distribution and its' application to activity duration distributions will be discussed in 6.4.13.

6.4.3 THE COMPOUND POISSON DISTRIBUTION

The Compound Poisson distribution as distribution for activity durations was suggested by Parks and Ramsing (1969).

Parks and Ramsing (1962: B-399) assume *there is a 100% probability that any project will take* *at least the minimum time to complete*. This leads them to the observation that the probability distribution will only extend from the minimum time out towards an undefined maximum. PERT, on the other hand, states that the *optimistic or minimum time must occur only one time in a hundred.*

To obtain the mean of the Poisson distribution for each activity, the minimum time is subtracted from the average time. The mean is then "tacked on" to the minimum time. The two subjective estimates needed as input are thus

1) the minimum time to complete a activity and
2) the average amount of time (arithmetic mean) to complete the activity.

This information can be used to determine the probability of completing a particular path by the scheduled completion date. A detailed description and example can be found in Parks and Ramsing (1969).

Parks and Ramsing (1969: B-402) note that, with a small number of activities, the use of *large size arrivals tends to give a lumpy distribution because of the discrete characteristics of the Poisson.* If the number of activities is large, a smooth distribution usually results.

Parks and Ramsing (1969) conclude that the decision on using the Poisson should be based on empirical data. They add that the existence of the Adelson's formula that can be applied to large networks to compute the probabilities involved to determine a criticality index more effectively is an additional factor that supports the decision to use the Poisson distribution. Parks and Ramsing (1969) argue that the use of the compound Poisson distribution with Adelson's formula offers a more cost effective way to determine the information contained in the criticality index than Monte Carlo simulation suggested by Van Slyke (1963) [reported in Parks & Ramsing (1969)].

The probability density function (Sichel 1975) is:

$$\phi(r) = \frac{((1-\theta)^{\frac{1}{2}})^{\gamma}(\frac{\alpha\theta}{2})^{r}}{K_{y}(\alpha(1-\theta)^{\frac{1}{2}})r!} \; K_{r+y}(\alpha)$$

where $r \geq 0$; $-\infty < y < \infty$; $0 < \theta < 1$; $\alpha > 0$; *$K_{\gamma}(\cdot)$ is the modified Besell function of the second kind of order* γ

Adelson's formula (Parks & Ramsing 1969: B-398) is:

$$R_{j+1} = \frac{1}{(j+1)}[a_1R_j + 2a_2R_{j-1} + 3a_3R_{j-2} + \ldots + (j+1)a_{j+1}R_0] \qquad where \quad R_0 = exp(-\sum_i a_i)$$

$R_j + 1$ = the density of the compound Poisson at $j+1$

a_j = the mean arrivals of a simple Poisson distribution with arrival size j

i = the time interval

6.4.4 THE UNIFORM AND TRIANGULAR DISTRIBUTION

The triangular distribution as distribution for activity durations was originally suggested by MacCrimmon and Ryavec (1964). The mean and standard deviation can be determined exactly.

In his discussion on probabilistic considerations pertaining to the PERT model, Elmaghraby (1977: 230) gives an example of how the uniform (when a and b represent the range of possible values and all the values between a and b are equally probable) and the triangular distribution (if we have three time estimates as in the case of the PERT model) can be applied as activity duration distributions.

The triangular distribution is also suggested by Williams (1992). He indicates that it is a generally accepted and easily understood distribution for project planners.

The expressions for the probability density function, mean and standard deviation in each case are:

1. **The uniform distribution**

$f(x) = 1/(b-a) \qquad a \leq x \leq b$

$E(Y) = (a + b)/2$

and $var\,(Y) = (b-a)^2/12$

2. **The triangular distribution**

$$\begin{aligned} f(x) &= \frac{2x}{bc} \quad for \quad 0 \leq x \leq b \quad (minimum\ zero) \\ &= \frac{2(c-x)}{c(c-b)} \quad\quad\quad b \leq x \leq c \\ E(Y) &= \frac{(b+c)}{3} \\ var(Y) &= \frac{(b^2-bc+c^2)}{18} \end{aligned}$$

$Mode = b$

The 10% and 90% points are $\sqrt{0.1mb}$ *and* $\sqrt{0.1b(b-m)}$ respectively (Williams 1992).

6.4.5 THE NORMAL DISTRIBUTION

The normal distribution is suggested as activity duration distribution by Sculli (1983) and Kamburowski (1985). Sculli (1983: 157) justifies his use of the normal distribution as follows: *most large networks can be reduced to a guide network, where a completely independent path becomes one activity. The central limit theorem justifies the Normality assumption for the duration of activities in the guide network.* Kamburowski (1985: 1057) claims that the simplicity of assuming normality in activity durations is valuable when *project network structure and evaluations of activity times may change* often during the project lifespan.

The probability density function is:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$$

where μ is the mean and σ^2 is the variance.

However, it is recognized in the literature that the distribution for activity durations is asymmetric and always positive. Using the normal distribution only in the positive region will result in a distribution that is unstable with respect to convolution (Golenko-Ginzburg 1989).

6.4.6 THE BERNY DISTRIBUTION

Berny (1989) considers the ability of people to provide valid information as central to the need to reconsider the underlying assumptions for generating a new distribution for activity durations.

Berny (1989: 1121) consulted with project managers and advises that the following three estimates are practical and realistic to make:

1) the most likely value (mode)
2) the shortest time or lowest cost. If it is not available, the *lowest limit with an associated probability is suggested.*
3) an *estimate of chance to exceed the mode.* A limited choice can be given, as this is a difficult estimate. Values of high (75%), medium (60%) and low(45%) are suggested.

Berny (1989) proposes a growth curve model to assess risk.

The probability density function is given by

$$\frac{dP}{dx} = \left(\frac{P_M}{x_M^{m}}\right)(m-1)x^{m-1}\exp\left[(1-\frac{1}{m})(1-\frac{x}{x_M})^m\right]$$

where $m > 1$, $x_M = T_M - T_0$ is the scaled mode and T_M is the actual mode, hence $x_M > 0$, and P_M is the probability to exceed the mode (Berny 1989: 1123).

The distribution is comprehensively described in Berny (1989) and is incorporated in a module of VISIER, a software package that has been developed by Berny.

Berny (1989) mentions the following advantages of the proposed risk function: it allows control of the lower limit; it does not depend on an upper limit and the parameters can be determined exactly. However, the expected value and variance need to be determined iteratively.

6.4.7 A DISTRIBUTION SUGGESTED BY GOLENKO-GINZBURG

Golenko-Ginzburg (1989) suggests an asymmetric activity-time distribution which is stable with respect to maximization and convolution. He argues that the main problem with a number of activity time distributions that have been suggested in the literature is the non-stability with respect to convolution and maximization. An activity-time distribution is *unstable with respect to convolution (maximization) if the sum (maximum) of two independent activity-times has another distribution* (Golenko-Ginzburg 1989: 389).

Golenko-Ginzburg (1989) proposes the use of the cumulative distribution function:

$$F_v(x) = exp\,[-(\theta/x)^v] \qquad 0 < x < \infty$$

The value of θ can be determined through the relationship

$$m = \theta\left(\frac{v}{v+1}\right)^{\frac{1}{v}}$$

where m is the mode for a particular activity and v is the level of uncertainty for the total project which is kept constant. Golenko-Ginzburg (1989) advises a value of $v=2$.

The distribution is particularly useful when only one value is estimated, that is the most likely value. It is applicable in research and development projects where similar previous projects rarely exist, thus making "good" estimates of optimistic and pessimistic times very difficult.

6.4.8 THE PEARSON FAMILY OF DISTRIBUTIONS

Formulae for the mean and standard deviation of random variables from judgemental estimates that have been developed by Pearson and Tukey (1965) for the Pearson family of distributions have been found to be more accurate than their competitors (Keefer & Bodily 1983). Ranasinghe (1994) suggests these formulae as the most suitable to use for generating the moments of the distribution for activity durations. He based the use of this group of distributions on the ability of the distributions to reflect skewness and peakedness. Lau and Somarajan (1995: 45) also remark on the fact that this group of distributions can *collectively model all the possible combinations of the four main distribution characteristics.*

The formulae are:

$$E(X) = P_{0.5} + 0.185\Delta \qquad (1)$$

where $\Delta = P_{0.95} + P_{0.05} - 2P_{0.5}$

$$\textit{and} \quad \sigma = \frac{P_{0.95} - P_{0.05}}{\max\left[3.29 - 0.1(\frac{\Delta}{\sigma^*})^2, 3.08\right]}$$

where $\sigma^* = \dfrac{P_{0.95} - P_{0.5}}{3.25}$ and P_x is the x-th percentile value.

6.4.9 THE GAMMA DISTRIBUTION

The gamma distribution was suggested by Williams (1992). His argument for using the distribution is that it can be used in cases where an upper limit is specifically not wanted.

The probability density function is

$$f(x) = \frac{\left(\frac{x}{b}\right)^{c-1} e^{-xb}}{\Gamma(c) \cdot b} \qquad for \quad x \geq 0$$

with mean: bc

variance: b^2c

and mode: $b(c\text{-}1)$

Calculation of incomplete gamma functions is required to determine the 10% and 90% points (Williams 1992: 270).

To fit a gamma distribution with minimum A and parameters b and c, given a mean μ, variance σ^2 and mode M (Williams 1992: 270), the following relationships can be used:

$b = \mu - M$

$c = \sigma^2 / b^2$

$A = \mu - bc$

6.4.10 THE MULTINOMIAL DISTRIBUTION

The multinomial distribution as distribution for activity time durations, specifically applied to information systems and design projects, was suggested by Bonett and Deckro (1993).

Bonett and Deckro (1993) argue that activity completion times (i.e. activity durations) are generally estimated as discrete time values, e.g. it will take 3 or 4 months. They suggest that a database on completed projects, as well as information obtained from "experts", can be used to assign probabilities to each of the estimated discrete time values. By using the multinomial assumption, the mean expected completion time for each activity (task) can then be defined as

$$\mu_i = \sum_j \; [\pi_{ij} t_{ij}]$$

where π_{ij} is the probability of completing task i in time category j and t_{ij} is the duration of task i in category j.

The mean completion times for all the activities are then used to determine the critical path. Bonett and Deckro (1993) derive the expressions for the estimated mean project duration as

$$\mu = \sum_i [t_i \pi_i]$$

and the associated estimated variance as

$$\sigma^2 = \sum_i [(t_i - \mu)^2 \pi_i]$$

where π_i is the probability associated with a given value of the project duration t_i.

The probability to complete the project on or before a specified duration can now be obtained directly from the cumulative probabilities (Bonett & Deckro 1993).

The methodology of Bonett and Deckro is referred to as multinomial PERT (M-PERT). They argue that their approach captures the expert's estimates of the likelihood for each activity completion time more accurately and do not force the time estimates into a specific approximation for the Beta distribution. They conclude that the Beta distribution, being a continuous distribution, implies that any fraction of a time period can be accurately estimated and assigned to a task, an aspect not typical to the estimation of durations of activities. The assumption of normality for project duration is another concern as it does not hold if the number of activities on the critical path is small (Bonett & Deckro 1993).

The drawback of this approach lies, once again, in the collection of the data required. The estimates, both for the time of an activity and its associated probability, are still subjective and difficult to obtain. A formal approach to elicit the time and probability estimates are needed before comparisons can be made with other approaches. However, it offers a new venue for research, i.e. is the distribution of activity time duration discrete or continuous? If discrete, what is the best way to obtain the time estimates and probabilities involved?

6.4.11 THE LOGNORMAL DISTRIBUTION

Ranasinghe (1994) uses the two parameter lognormal distribution for activity durations when discussing the quantification and management of uncertainty in activity duration networks. He bases the use of this distribution on the fact that it can only assume positive values and is also positively skewed, two important characteristics of activity durations.

The mean and variance of the lognormal distribution is given by

$$\mu = me^{\frac{1}{2}\sigma^2}$$

$$\text{and} \quad \sigma^2 = m^2w(w-1)$$

where m is the median and $w = e^{\sigma 2}$ (Hasting & Peacock 1974).

6.4.12 THE ERLANG DISTRIBUTION

The Erlang distribution, as distribution for activity time durations, is suggested by Bendell et al. (1995). The parameters of the Erlang distribution can be chosen so as to provide a good fit to most unimodal activity time distributions. It was chosen on the grounds of three selection criteria, which are:

1) Expressions for the first four central moments of the distribution of the maximum and the sum of two or more activity times need to be easy to derive.
2) Choose an asymmetric distribution to reflect the actual properties of activity times. MacCrimmon and Ryavec (1964: 20) suggest that the actual distribution of activity durations have three properties: *unimodality, continuity and two nonnegative abscissa intercepts.*
3) The distribution of the maximum and sum of two activity times should be of the same type as that of the individual activity times.

The probability density function for the Erlang distribution is:

$$f_1 = \frac{\lambda_1^{c_1} x_1^{c_1-1} e^{(-\lambda_1 x)}}{(c_1-1)!} \qquad 0 < x < \infty$$

The cumulative distribution function is

$$F_1 = 1 - e^{(-\lambda_1 x)}[\sum_{i=0}^{c_i-1} (\frac{\lambda_1 x}{i})^i]$$

$$where \quad \lambda_1 = \frac{c_1}{\mu_1} = \frac{Shape\ parameter}{mean}$$

The input parameters for the Erlang distribution can be obtained from the three time estimates *a,m* and *b* described in 6.4.2. above. They are

$$\mu = \frac{(a+4m+b)}{6} \quad and \quad C = \left(\frac{a+4m+b}{b-a}\right)^2$$

Bendell et al. (1995) describe, by means of an example, the estimation of the parameters of the Erlang distribution if sample data is available. In his example the activity time distributions were varied (it includes unimodal and J-shaped distributions).

Bendell et al. (1995) cite that information regarding the activity times may sometimes suggest bimodality. It could happen when the data comes from two different sources, or if it depicts two different scenarios for an activity, e.g. the activity time in the case of fine or foul weather.

The probability distribution for bimodal activity time can be formed by taking a weighted average of two Erlang probability distribution functions (Bendell et al. 1995: 875), i.e.

$$\pi f(t_1; \lambda_1, C_1) + (1 - \pi) f(t_2 ; \lambda_2 , C_2)$$

where π is the weighting factor.

Bendell et al. (1995) conclude that the Erlang distribution is useful when activity times have skew distributions or where activity duration distributions are in empirical form. It also requires a minimum of only two input parameters for each activity, making it an attractive option in real-life applications.

6.4.13 THE RAMBERG-SCHMEISER DISTRIBUTION

The Ramberg-Schmeiser distribution, as a supplement to the Beta distribution for activity time distribution, was proposed by Lau and Somarajan (1995).

The skewness-kurtosis (β_1, β_2) diagram is usually used to compare the ability of distributions to handle different shapes. Skewness (β_1) and kurtosis (β_2) are defined as

$$\beta_1(x) = \frac{\mu_3(x)}{[\mu_2(x)]^{1.5}}$$

$$\beta_2(x) = \frac{\mu_4(x)}{[\mu_2(x)]^2}$$

where μ_x is the expected or mean value

and $\mu_k(x)$ is the k-th central moment.

The Beta distribution does not cover the entire possible (β_1, β_2) area.

Lau and Somarajan (1995: 46) choose the Ramberg-Schmeiser distribution to complement the Beta distribution on the following grounds:

1) *it has a closed-form inverse cdf* (cumulative distribution function) *with parameters (a,b,c,d):*

$$R(p) = F_R^{-1}(t) = a + [p^c - (1-p)^d]/b \qquad 0 \leq p \leq 1$$

2) *the closed-form inverse cdf makes it very easy to generate random variates for simulation using the inverse transform method*

3) it complements the β-distribution regarding coverage of the (β_1, β_2) area.

The distribution is described in Ramberg and Schmeiser (1974).

6.5 A COMPARATIVE STUDY OF ACTIVITY DURATION DISTRIBUTIONS

6.5.1 COMPARATIVE TABLE OF INDICATORS FOR SUGGESTED DISTRIBUTIONS

DISTRIBUTION	DISCRETE/ CONTINUOUS	NUMBER OF PARAMETERS	ESTIMATES REQUIRED FOR EACH ACTIVITY in PERT	EXACT/APPROXIMATE DETERMINATION OF MEAN AND STANDARD DEVIATION
BETA	CONTINUOUS	4 (α; γ; a, b)	3 (a, b, m)	AN APPROXIMATION USED IN PERT
COMPOUND POISSON	DISCRETE	3 (θ; α; γ)	2 (minimum time, arithmetic mean)	EXACT
UNIFORM	CONTINUOUS	2 (a, b)	2 (a, b)	EXACT
TRIANGULAR	CONTINUOUS	2 (b, c)	3 (a, b, m)	EXACT
NORMAL	CONTINUOUS	2 (μ, σ)	not indicated	APPROXIMATION
BERNY	CONTINUOUS	4 T_M, T_0; P_M, P(0)	3 (minimum duration, mode, P_M)	ITERATIVELY - (The parameters are exactly determined)
GOLENKO-GINZBURG	CONTINUOUS	2 (θ, v) [θ varies, v is kept constant)	1 - mode (the value of v is pre-given).	not indicated
PEARSON FAMILY	CONTINUOUS	4 (the maximum number of parameters for this family of distributions)	3 (if using the approach suggested in Ranasinghe (1994)).	Approximation used in PERT
GAMMA	CONTINUOUS	2 (b, c)	3	EXACT
MULTINOMIAL	DISCRETE	n (number of possible discrete time values for each activity)	the possible discrete time values for each task and their associated probabilities	EXACT
TWO-PARAMETER LOGNORMAL	CONTINUOUS	2 (μ; σ^2) (as defined in 6.4.11)	3 (using the approach of Ranansinghe (1994)).	APPROXIMATION
ERLANG	CONTINUOUS	2 (λ, c)	A minimum of 3	EXACT
RAMBERG-SCHMEISER	CONTINUOUS	4 (a, b, c, d)	A minimum of 7 fractiles is suggested	APPROXIMATION

Table 6.1 Comparative table of indicators for suggested distributions

6.5.2 ADVANTAGES/DISADVANTAGES OF ACTIVITY DURATION DISTRIBUTIONS

DISTRIBUTION	ADVANTAGES	DISADVANTAGES
Beta	Ease of use in practise. Flexibility, finite range possible.	Shape is restricted in PERT. Does not cover all possible skewness and kurtosis values.
Compound Poisson	Mathematical ease by using Adelson's formula. This will probably no longer be a determining factor due to the increase in computing power.	Usefulness in smaller networks limited. Adaption of PERT assumptions must be acknowledged.
Uniform	Easy to use	Limited application in practise
Triangular	Easy to use and understand	The limits required can be an unreasonable assumption.
Normal	Ease of use. Lead to natural extension for distribution of project completion time.	The normal distribution is symmetric while the nature of activity durations requires an asymmetric distribution
Berny	Allows control of lower limit. It does not depend on an upper limit.	Iterative determination of mean and variance. Lacks visibility to project planner, e.g. what is the effect of changing the "probability of exceeding the mode"?
Golenko-Ginzburg	Only one subjective estimate (the mode) needed. Particularly useful for entirely new types of projects.	
Pearson Family	Covers a wide range of possible distributions for activity durations as special cases. Ability to reflect skewness and peakedness. Formulae from Pearson and Tukey are considered most suitable to use to generate moments.	
Gamma	Suitable when no upper limit is specifically wanted.	Calculations not easy. Require tables or computer algorithms.
Multinomial	Applicable when number of activities on critical path are small. Suit the way people estimate time for activities, e.g. 3 weeks, 2 days, one year etc.	The elicitation of the subjective time estimates and their associated probability.
Two-parameter lognormal	Describe important characteristics of activity durations	
Erlang	Only two time estimates needed. Can accommodate bimodality.	
Ramberg-Schmeisser	Can be used to complement the Beta-distribution to cover all (β_1, β_2) values.	Computations fairly difficult, but software exists to handle the computations.

Table 6.2 Advantages/disadvantages of activity duration distributions

6.5.3 ACTIVITY DURATIONS DISTRIBUTIONS: A SUMMARY

The existence of many different types of activities indicates that there will be as many distributions to fit to the different categories of activities. This viewpoint is supported by the wide variety of distributions suggested in the literature. However, the reasons for choosing the distribution were, in most cases, not linked to the type of activity but to certain mathematical prerequisites.

In PERT analysis, the expected time and standard deviation of an activity is required. The initial suggested distribution to obtain the above parameters, the Beta distribution, has been extensively studied and criticised as an activity duration distribution as pointed out in 6.4.2. The Beta distribution, has, however, in practise and in project management software packages, remained the standard distribution to use. The importance given to the distribution in industry is probably due to the fact that the Beta distribution offers a trade-off between mathematical correctness and practical use.

The compound Poisson distribution, suggested by Parks and Ramsing (1969), was mainly chosen on the grounds of the existence of a formula to compute the criticality index more cost-efficient at that time. This argument does not hold any more, given the power of current computer technology. However, activities that exhibit the type of behaviour that can be characterized by the compound Poisson distribution, described in Parks and Ramsing (1969), can be analysed using this distribution. When using this distribution, the adaption of the PERT assumptions must be borne in mind.

The uniform distribution will have only limited use in practise. It requires only a minimum and maximum value for the time of each activity (Williams 1992) whereas the most important estimate for activity durations is the position parameter, i.e. the mean, mode or median.

Williams (1992) emphasizes practicality and ease of use when suggesting the triangular distribution as an activity duration distribution. It is also an attractive alternative because the same three initial subjective estimates required by the Beta distribution can be used as input to the triangular distribution. Furthermore, the mean and standard deviation can be determined exactly. Williams (1992) has found that project planners have positively accepted this distribution.

The normal distribution was mainly used to be able to comply to certain mathematical properties of the completion time distribution. It is not a good distribution to represent activity durations as it is symmetric and cover both positive and negative values.

The growth curve function, suggested by Berny (1989), was mainly proposed to benefit from those subjective estimates that people might be more familiar with and thus will give more valid results.

The non-stability, of distributions suggested for activity durations, with respect to convolution and maximization, was the principal reason behind the suggestion of a new distribution by Golenko-Ginzburg (1989). This distribution is stable with respect to convolution and maximization. An added advantage is that only one input value, the mode, is required. As stated earlier, subjective estimates of central fractiles are also more reliable than subjective estimates for the extreme fractiles.

Ranasinghe (1994) and Lau and Somarajan (1995) propose the encompassing Pearson Family of distributions. Lau and Somarajan (1995) indicate that it collectively covers all the possible combinations of the four parameters that characterize a distribution. Ranasinghe (1994) notes the ability of the Pearson family of distributions to reflect the skewness and peakedness of activity durations. The existence and accuracy of the approximation formulae for the mean and standard deviation of random variables from judgemental estimates that has been developed by Pearson and Tukey (1965) for the Pearson family of distributions is another motivation for using this family of distributions.

The only reference to the gamma distribution as distribution for activity durations was found in Williams (1992). It was suggested as an alternative to the triangular distribution if one do not want an upper limit for the duration of the activities under consideration.

Bonett and Deckro (1993) suggested a discrete distribution, the multinomial. They based the use of this distribution on two premises, namely:

1) when the number of activities on the critical path is small, the normal distribution cannot

be used for the project duration distribution.

2) that judgemental estimates are discrete time values. They also argue that if a continuous distribution is assumed, any fraction of a time period can be accurately estimated and assigned to a task, an aspect not typical to the estimation of durations of activities.

The main disadvantage to this approach lies once again in obtaining valid time estimates and their associated probabilities. No formal approach to obtain these has been suggested.

The lognormal distribution has been used as an activity duration distribution by Ranasinghe (1994). He based his argument on the fact that the lognormal distribution exhibits two important characteristics of activity durations, i.e it can only assume positive values and is positively skewed.

Bendell et al. (1995) suggest the Erlang distribution. They based their decision on three aspects described in 6.4. This approach is useful when activity times have skew distributions or where the data is in empirical form. It also requires a minimum of only two input parameters for each activity, a practical advantage.

The Ramberg-Schmeiser distribution was suggested as a complementary distribution to the Beta distribution to describe activity durations (Lau & Somarajan 1995). The reasons for choosing this particular distribution are provided in 6.4.13 above.

6.5.4 CONCLUSION

Based on the review of distributions suggested in the literature, one is inclined to agree that no single distribution is universally applicable.

It is very difficult to determine theoretically the "goodness" of the fit of a mathematical distribution for activity durations. Poor fit, according to Lau and Somarajan (1995) can be due to

1) inaccurate estimates by the expert or
2) the selection of an inappropriate distribution function.

Furthermore, in choosing an activity duration distribution, a trade-off is usually sought between mathematical correctness and practical userfriendliness.

The key criteria for the selection of an appropriate distribution for activity durations are identified in the next section.

6.6 SELECTION CRITERIA FOR ACTIVITY DURATION DISTRIBUTIONS

Several authors have indicated their specific criteria for selecting a distribution.

Williams (1992) notes that information regarding the parameters of position, spread and skewness as well as the minimum and maximum values needs to be specified in order to define an activity duration distribution.

Bendell et al. (1995) cite three aspects that need to be considered when choosing a distribution for activity durations in 6.4.12.

Lau and Somarajan (1995) mentioned two mathematical criteria in 6.4.13.

Regarding the balance between mathematical accuracy and practical use when choosing a distribution, Williams (1992: 265) remarks: *It is important that the distributions and parameters used are sufficiently flexible and facilitate elicitation from experts while not involving such mathematical complexity that they interfere with the more fundamental requirements to assess, analyse, monitor and manage project risk.*

Different criteria is thus applied when choosing an appropriate distribution for activity durations.

The determination of a statistical distribution of activity durations is thus influenced by:

1) the sample information available. This is usually in the form of subjective estimates made by an "expert".
2) the role of the activity duration distribution in determining the project completion time

distribution. The distribution of the sum of activity duration distributions needs to be considered when choosing an appropriate distribution.

3) the activity network framework, i.e. the role of the activity duration distribution within activity networks has to be clearly defined.

4) the intrinsic properties of activity durations. This include

   a) positive values *($t > 0$)*;

   b) skewness or bimodality

   c) the discrete or continuous nature of activity durations.

These four aspects will ultimately determine the selection of an appropriate distribution and are depicted in figure 6.1.

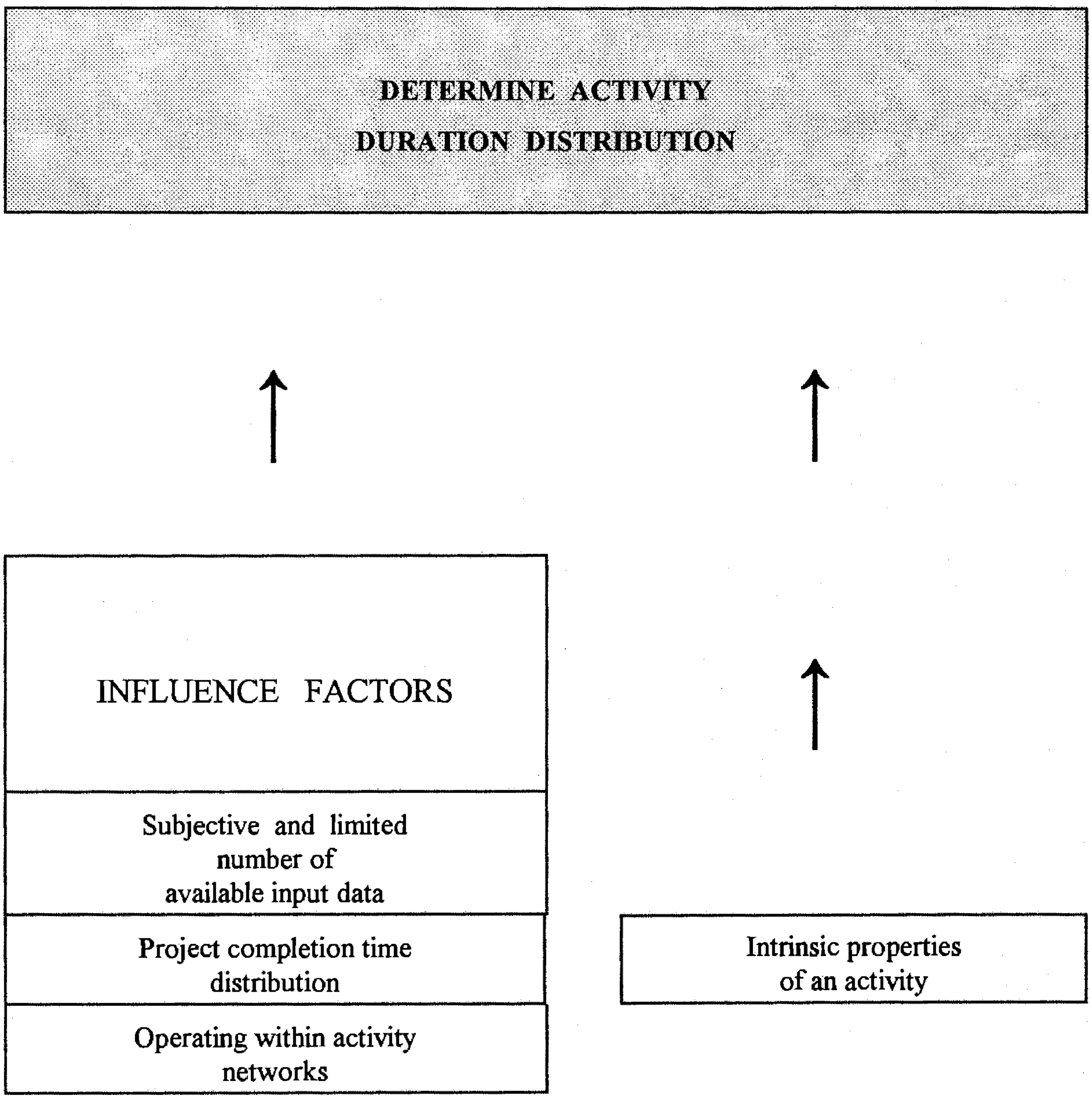

DETERMINE ACTIVITY
DURATION DISTRIBUTION
INFLUENCE FACTORS
Subjective and limited number of available input data
Project completion time distribution
Operating within activity networks
Intrinsic properties of an activity

Figure 6.1 Selection criteria for determining activity duration distribution

6.7 A REVIEW OF ESTIMATION ISSUES RELATING TO ACTIVITY DURATIONS

Numerous works have been published on the estimation of PERT activity time parameters and issues related to activity-based inaccuracies. The completion time of the predetermined critical path (and thus the project completion time) in the PERT method is determined by these parameters and thus they are of cardinal importance. References cited here are those considered by the author to be relevant to the contents of the chapter.

Van Slyke (1963) uses Monte Carlo simulations to provide unbiased statistical estimates for the mean and standard deviation required in PERT analysis.

Moder and Rodgers (1968) study the aspect of estimating the moments of a statistical distribution from judgement estimates of various percentiles of the distribution and its mode. They examined five distributions: the normal, beta, triangular, uniform and exponential. They found that the 5th and 95th percentiles are superior, with regard to robustness to variations in the shape of the distribution, to the 0 (a) and 100 (b) percentiles used in the classic version of PERT. In their work, they recommend the following formulae for μ and σ.

$$\hat{\mu} = \frac{(a_5 + 4m + b_{95})}{6} \quad \textit{and} \quad \hat{\sigma}^2 = \frac{(b_{95} - a_5)^2}{10.2}$$

where a_5 and b_{95} refers to the 5-th and 95-th percentile respectively and m is the mode.

Moder and Rodgers (1968) also recommend the comparison of the actual and estimated values by means of a quality control chart. The chart is based on examining the normalized error that is obtained by dividing the difference between the actual and estimated activity duration time by the estimated standard deviation. Such comparisons can lead to meaningful insight into the estimation process and serves as a calibration instrument. However, as in the case of software cost estimation, the lack of a database on historical projects is a common industry phenomena. Post mortems on past projects are rare events. It does not seem as if they want to learn from the past!

Britney (1976) suggests that the cost of under-and overestimation of activity duration times is rarely acknowledged. Costly project delays can be the result of underestimation and overestimation can result in inactivity, also causing significant losses.

Britney (1976) formulates Bayes point estimates for Beta-distributed activity times by minimising the potential loss of misestimation. The Bayesian point estimate for the mean is viewed as a certainty equivalent. Britney's procedure is called BPERT (Bayesian PERT). It focuses on the loss aspect of misestimation and provides *optimal deterministic equivalent statistics for stochastically distributed variables* (Britney 1976: 939).

Britney (1976) concludes that, in the case of positively skewed distributions linked to loss ratios greater than unity, the point estimates for activity durations will be significantly larger than the mean. BPERT will produce less optimistic completion dates than conventional PERT for projects with these characteristics.

As the classical PERT approach do not address the economics of estimation, BPERT provides an alternative.

Littlefield and Randolph (1987) mention that the expression for the mean

$$\mu = \frac{a + 4m + b}{6}$$

depend on the following assumptions:

1) That the distribution for activity duration times is the Beta-distribution.
2) That the judgemental estimation of *a, b,* and *m* are done competently by the people involved.
3) That the standard deviation is one sixth of the range.
4) That the *linear approximation,*

$$\boldsymbol{Approx.}\ (Mean) = \frac{(1+4m)}{6}$$

is acceptable for estimating the mean.

Gallagher (1987: 1360) indicates that, when applying these formulae

$$\mu = \frac{a+4m+b}{6} \quad and \quad \sigma^2 = \left(\frac{b-a}{6}\right)^2$$

one can assume that the *standard deviation is exact and the mean approximate or vice versa.*

Farnum and Stanton (1987) show that the expression

$$\hat{\mu}_x = 4\hat{m}_x + 1 \quad (\hat{m}_x \ \ denotes\ the\ mode)$$

closely approximates the actual relationship between the mean and the mode for a large range of possible modal values but fails if the mode is outside the interval $a + 0.13(b\text{-}a) < \text{m} < b - 0.13(b\text{-}a)$. This happens if the standard deviation is much smaller than $(b\text{-}a)/6$. They propose the following alternative expressions for $\hat{\mu}_x$ *and* $\hat{\sigma}_x$ if the mode is near the upper or lower limits of the distribution.

$$For \quad \hat{m}_x < 0{,}13: \qquad \hat{\mu}_x = \frac{2}{2+\frac{1}{\hat{m}_x}} \quad and \quad \hat{\sigma}_x = \left[\frac{\hat{m}_x^2(1-\hat{m}_x)}{(1+\hat{m}_x)}\right]^{\frac{1}{2}}$$

$$For \quad \hat{m}_x > 0{,}87: \quad \hat{\mu}_x = \frac{1}{3-2\hat{m}_x} \quad and \quad \hat{\sigma}_x = \left[\frac{\hat{m}_x(1-\hat{m}_x)^2}{2-\hat{m}_x}\right]^{\frac{1}{2}}$$

Williams (1992) addresses the important issues regarding

1) the number of parameters that should be estimated
2) the use of default and generic information
3) which parameters need to be estimated as well as the estimation of these in practise.

Williams (1992) considers five practical aspects when addressing the first issue, namely the number of parameters that should be estimated. They are:

1) the status of the project, i.e in which stage of development is the project
2) the availability of data on similar, previous projects
3) the level to which planners are capable of estimating the parameters
4) the time available and
5) ease of analysis within the activity network modelling framework.

Secondly, Williams (1992: 266) notes that information obtained through

1) a *general appreciation of project activities* and
2) analysis of previous similar projects

can be utilized. He has observed, through practical work at the YARD company, that activity durations tend to have a skewness of 2:1 (the mode is 1/3 along the range). He suggests that this can be a useful default if no knowledge regarding the activities are known.

Another useful instrument is the categorisation of levels of uncertainty for different activity types (Williams 1992). Each level has a generic factor. The factor is defined as

$$\textit{Generic factor} = \frac{\textit{Standard Deviation}}{\textit{Duration Position Parameter}}$$

The duration position parameter can be the mean, mode or median.

An example from Williams (1992) will shed some light on the use of this technique:

UNCERTAINTY LEVEL		GENERIC ACTIVITY TYPE	GENERIC FACTOR
0	Deterministic		0
1	Low variability	Manufacturing	0.2
2	Medium variability	Full development	0.3
3	High variability	Project definition	0.4
4	Very high variability	Trials	0.6

Table 6.3 Categorisation of levels of uncertainty

In addressing the third issue, Williams (1992) argues that the position parameter is the most important parameter to estimate. It is also the one that project managers have the best possible "feel" for. However, it is important to determine if the judgmental estimate corresponds to the median, mode or mean.

Williams (1992) suggests the use of a set of parameters, consisting of:

1) the most likely value or the mean and
2) the probable limits (10% and 90% points) or the uncertainty level (described above). He argues that this is a natural set for managers to estimate at the early stages of a project.

Keefer and Verdini (1993) compare a number of proposed approximations, all based on the availability of only three judgmental probability assessments, for the mean and standard deviation of PERT activity times. The approximations used are shown in Table 6.4. Table 6.4 is an extract from Keefer and Verdini (1993: 1088).

Approximation	Approximation for Mean	Approximation for Variance
Original PERT formulas (A1)	$\hat{\mu} = [x(0.0) + 4x_m + x(1.0)]/6$	$\hat{\sigma}^2 = ([x(1.0) - x(0.0)]/6)^2$
Modified PERT Formulas (A2)	$\hat{\mu} = [x(0.01) + 4x_m + x(0.99)]/6$	$\hat{\sigma}^2 = ([x(0.99) - x(0.01)]/6)^2$
Extended Pearson-Tukey Approx. (A3)	$\hat{\mu} = 0.630x(0.50) + 0.185[x(0.05) + x(0.95)]$	$\hat{\sigma}^2 = 0.630[x(0.50) - \hat{\mu}]^2 + 0.185([x(0.05) - \hat{\mu}]^2 + [x(0.95) - \hat{\mu}]^2)$
Extended Swanson-Megill Approx. (A4)	$\hat{\mu} = 0.400x(0.50) + 0.300[x(0.10) + x(0.90)]$	$\hat{\sigma}^2 = 0.400[x(0.50) - \hat{\mu}]^2 + 0.300([x(0.10) - \hat{\mu}]^2 + [x(0.90) - \hat{\mu}]^2)$
Troutt Formula for Mean (A5)	$\hat{\mu} = [x(0.0) + 4x(0.50) + x(1.0)]/6$	Not applicable
Farnum-Stanton Formulas (A6)	$\hat{\mu} = 2/(2 + 1/x_m), \quad x_m < 0.13$	$\hat{\sigma}^2 = x_m^2(1 - x_m)/(1 + x_m), \quad x_m < 0.13$
Golenko-Ginzburg Formulas (A7)	$\hat{\mu} = [2x(0.0) + 9x_m + 2x(1.0)]/13$	$\hat{\sigma}^2 = \frac{(x(1.0)-x(0.0))^2}{1268}\left[22 + 81\frac{x_m - x(0.0)}{x(1.0) - x(0.0)} - 81\left(\frac{x_m - x(0.0)}{x(1.0) - x(0.0)}\right)^2\right]$

Table 6.4 Approximations compared by Keefer & Verdini (1993)

They found that the two sets (A3 AND A4) of three-point approximations are the most accurate for the mean and variance of PERT activity times under the assumption that the activity duration (times) are Beta-distributed. Further research is required to look at the trade-offs when one has to choose between the two alternatives (Keefer & Verdini 1993). It is also important to determine the accuracy of these approximations under the assumption of the other suggested distributions for activity durations.

According to Ranasinghe (1994), the generation of probabilistic moments that represent the best knowledge about the input data is the first step in the quantification process of uncertainty in activity durations. As the estimates involved in activity durations are judgemental estimates, Ranasinghe (1994) suggests that the formulae formulated by Pearson and Tukey (1965) are the most suitable to generate the moments for activity distributions. The formulae were given in 6.4.8.

Lau and Somarajan (1995: 40) argue that the most common and straightforward method for estimating the cumulative distribution function of activity durations is the fractile method. They describe the method as follows:

Specify a number of (say n) required fractiles α_i's $(i = 1,2,...,n)$, elicit the corresponding time estimates t_i's. For example, if one of the α_i's is (say) $\alpha_3 = 0.4$, then ask the expert to estimate the magnitude of the target time t_3 such that the probability of T not exceeding t_3 is $\alpha_3 = 0.4$. Or more briefly, one estimates $T_{0.4}$.

Lau and Somarajan (1995) see PERT as a "poorly defined" fractile method because:

1) there is confusion in the literature as to what fractile *a* and *b* corresponds to, i.e. are they T_0 and T_1, $T_{0.01}$ and $T_{0..99}$ or $T_{0.05}$ and $T_{0.95}$? A prerequisite for the formula

$$\mu = \frac{a+4m\ b}{6}$$

is that *a* and *b* corresponds to T_0 and T_1 (Littlefield & Randolph 1987)

2) estimating *m* (the mode) is not estimating a fractile. Lau and Somarajan warns that a person making the estimate may confuse the median and the mode.

In using the fractile method, it is important to decide on the number of fractiles, which specific fractiles are to be used and the order in which they will be estimated (Lau & Somarajan 1995).

Lau and Somarajan (1995: 42) suggest the use of the following fractile procedure (Selvidge 1980) to estimate stochastic activity durations. The procedure is as follows:

1) Assess seven fractiles. That is, the three central fractiles: the 0.25, 0.50 and 0.75 *fractiles; and the four extreme fractiles: the 0.01, 0.10, 0.90 and 0.99 fractiles.*

2) *Assess the central fractiles first.*

They also cited several studies that confirm that people can estimate central fractiles more accurately than extreme fractiles.

Lau and Somarajan (1995) show the fitting of the fractiles to a Beta distribution. This results in exact expressions for u and o, while approximations are used in PERT.

It is clear from the above that the last word has not been spoken regarding the estimation of duration times in activity networks. The question remains: What form of estimate will produce estimates that are reliable, accurate and practical to make?

By approaching the problem from a distribution free viewpoint, the aspect of choosing the "correct" distribution may be eliminated.

6.7.1 DISTRIBUTION-FREE APPROXIMATIONS

Keefer (1994: 761) defines an n-point discrete approximation as follows:

An n-point discrete-distribution approximation consists of n values $x_1,...,x_n$ and corresponding probabilities of occurrence $p(x_1), p(x_2),...,p(x_n)$ chosen to approximate the probability distribution function of the underlying continuous random variable X.

Perry and Greig (1975) argue that the underlying distribution can be ignored when applying the following three-point-approximations to estimate the mean and variance of subjective probability distributions. They are

$$\sigma = \frac{(p_{95} - p_5)}{d} \qquad \textit{where} \qquad d = 3.25$$

and

$$\mu = \frac{(p_5 + 0.95m + p_{95})}{2.95}$$

The formula

$$\mu = p_{50} + 0.185\ (p_{95} + p_5 - 2p_{50})$$

using the median instead of the mode, suggested by Pearson and Tukey (1965), is also regarded as an accurate and distribution-free formula for the mean by Perry and Greig (1975).

Keefer and Bodily (1983) suggest that three-point approximations represent smooth unimodal probability distributions that are not extremely skewed or peaked, making it useful for judgemental assessments. Keefer (1994) regards the three-point discrete-distribution approximations for continuous probability distributions as distribution-free (the type of probability distribution does not have to be known).

The exact extent of the implications of distribution-free approximations for the mean and variance of activity durations on the project completion time distribution requires additional research.

6.8 CURRENT RESEARCH

Three important research directions are briefly described.

6.8.1 THE PM-NET AND BSPN MODELS

A brief summary of the most recent suggested models for dealing with software project management within the network contents, PM-Net and BSPN, is provided.

1 PM-Net: a software project management representation model

Lee et al. (1994) propose the PM-Net model. The model concentrates on software project control.

Lee et al. (1994) list five criteria an ideal model should adhere to:

1) As software development is a design process, this should be adequately described by the model.
2) In a software project, it is possible to start an activity even before all its prior activities are completed. This should be reflected in the model.
3) Changing requirements implies that the model should be able to indicate affected activities and resources, as well as the condition of these activities, i.e whether they are to re-executed or to be suspended.
4) The criteria that trigger the start of an activity should be included.
5) Information regarding the budget should be included in the model.

The PM-Net model is an extended and modified version of DesignNet.
The DesignNet model is described in Liu and Horowitz (1989).

To address the requirements of managers at all levels of the organisation, Lee et al. (1994) adopted the Data Flow Diagram (DFD) technique instead of the waterfall model that was used in DesignNet. The DFD technique allows for the decomposition of the project into distinct processes, then into activities, subactivities and eventually a set of tasks. Aspects that were modified from the DesignNet model were the transition firing rule, the token propagating rule and the token types.

The PM-Net design concepts such as the structure; the interconnection of activities; the token state types and token type notation; token propagation and control status operator propagation; the enabling and firing of a transition whenever an event occurs and the priority of firing a transition are discussed in Lee et al. (1994).

PM-Net provides a flexible representative method for different requirements, regarding the software control process, by different levels of managers and this is regarded as the biggest advantage of the model (Lee et al. 1994).

Lee et al. (1994) emphasize that PM-Net is a model for the representation and control of the rate of progress of a currently executing software project. It can describe and monitor the software development process. They stress the reliance of their model on an integrated database. The model can be applied in the planning stage to establish the activities and the relationships between activities.

2 A Beta-distributed Stochastic Petri Net (BSPN) model for software project time/cost management

Lee and Murata (1994) argue that a model that can simulate the behaviour of the project is needed by project managers to forecast and control project states. They propose the BSPN model, an integrated model of the program evaluation and review technique (PERT) and Petri nets, to address the time and cost aspects of a software project. The model inherits concepts and analysis methods from Petri nets (such as reachability, activity sequence and degree of concurrency) and concepts of time and cost management methods from PERT (Lee & Murata 1994).

Software project management is hierarchical in nature and is naturally modelled by the *folding and unfolding (or top-down step-wise refinement, divide-and-conquer method) concept of BSPN's* (Lee & Murata 1994: 152).

The stochastic transitions of a BSPN corresponds to an activity in a project. The duration (time) or cost of an activity in PERT is assumed to follow a Beta-distribution. In the BSPN model, the firing delays of the transition is assumed to follow a Beta-distribution. They are estimated by using the optimistic, pessimistic and most likely v times. Lee and Murata (1994) remark that a BSPN in which only the mean value of the delays is considered, can be analysed as a timed Petri net having deterministic time transitions but current analysis algorithms are not directly applicable to a BSPN.

Advantages of the BSPN model (Lee & Murata 1994: 164) are:

1) It is an *integrated, executable, and formal model.* It has the advantage of mathematical background, algorithms for analysis and software packages from PERT and Petri nets.

2) The *software project management WBS, software structure, development life cycle, and development team structure* is combined into an integrated BSPN structure.

3) The BSPN can concurrently model and analyze time and cost of a project.

4) The *BSPN can model and analyze all types of activity relations, uncertainty of activity duration and cost, and decisions* (or choice) in a project.

Problems encountered with the model are

1) The assumption of the Beta-distribution causes a statistical error.

2) The modelling power of a BSPN is high but the analysis is complex.

3) The modelling and analysis of manpower and development tools amongst the resources cannot be modelled by the proposed BSPN model.

6.8.2 Babu and Suresh (1996) develop optimization models to study time/cost/quality tradeoffs in project management quantitatively. This study is the first to consider quality as an additional trade-off variable.

6.8.3 MIPS, a decision support system (DSS) for interactive resource constrained project scheduling with multiple objectives, was developed by Rys, Stanek & Ziembla (1994). This system does not only solve the multiobjective project scheduling problem but also helps the user to select interactively the solution which he considers to be the best for his set of circumstances.

6.9 CONCLUSION

The urgent need for more accurate quantification of uncertainty in activity durations within activity networks is clearly demonstrated by the ongoing research on this topic, both from a theoretical as well as a practical perspective.

The subjective nature of the estimates for activity durations leads to wide variation due to personal and psychological factors and have to be acknowledged. As such, improvement must be sought through data collected on previous and current projects (a critical aspect also mentioned with regard to software cost estimation and software reliability in chapter 5) and the use of methods such as bootstrapping to improve the quality of the estimates.

The following four aspects were identified as the key aspects in determining the distribution that will be used:

1) the sample information available. This is usually in the form of subjective estimates made by an "expert".
2) mathematical prerequisites. The activity duration distribution influences the project completion time distribution. The distribution of the sum of activity durations needs to be considered when choosing an appropriate distribution.
3) the activity network framework, i. e. the role of the activity duration distribution within activity networks has to be clearly defined. One important aspect is the "mathematical easiness" of the distribution.
4) the intrinsic properties of activity durations. This include
   a) positive values *($t > 0$)*;
   b) skewness or bimodality and
   c) the discrete or continuous nature of activity durations.

In reviewing the statistical distributions that have been suggested to model the distribution of activity durations, the author comes to the conclusion that each distribution was chosen with a specific aim, that relates to either the type of projects considered or mathematical correctness, in mind.

Although criticised in the literature, the Beta distribution remains the standard and most popular choice in practise. Its main attractiveness lies in the fact that it offers a way of estimation that project managers can relate to. The triangular distribution is also often used as it offers the same degree of "easiness to understand" as the Beta. The uniform distribution, although easy to use, has limited use in practise as it is not practical to work with only an estimated minimum and maximum duration. Furthermore, as noted before, the estimation of endpoints are much more difficult than points in the centre.

The Compound Poisson, gamma, normal, Erlang, Ramberg-Shmeisser and the distribution suggested by Golenko-Ginzburg (1989) were all chosen to address mainly mathematical criteria related to activity networks.

The use of the multinomial distribution is, except for the use of the Compound Poisson (which was suggested specifically to address the computation aspect of the criticality index), the only discrete distribution suggested in the literature. As empirical judgmental estimates are discrete, it seems logical to do more research on the applicability and consequences of using discrete distributions in activity networks.

The Pearson family of distributions is a strong candidate as it addresses all four selection criteria mentioned above. Both the Gamma and Beta distributions are included in this family of distributions.

The lognormal distribution does represent the important characteristics of an activity duration very well, i.e. it can only assume positive values and the distribution is usually skew. More research is required to establish the advantages and disadvantages of using the lognormal distribution as activity duration distribution.

Research is also required to establish the project completion distribution if we use the "so-called" distribution-free formula for the mean and variance of activity durations.

Perhaps, Aristotle [Putnam & Fitzsimmons 1979: 194] should have the last word in this regard: *It is the mark of an instructed mind to rest satisfied with the degree of precision which the nature of the subject admits and not to seek exactness when only an approximation of the truth is possible.*

REFERENCES

Abdel-Hamid, T.K. 1990. On the Utility of Historical Project Statistics for Cost and Schedule Estimation: Results from a Simulation-based Case Study. *Journal of Systems and Software*, 13:71-78.

Abdel-Hamid, T.K. 1993. A Multiproject Perspective of Single-Project Dynamics. *Journal of Systems and Software*, 22(3):151-165.

Adlakha, V.G. & Kulkarni, V.G. 1989. A CLASSIFIED BIBLIOGRAPHY OF RESEARCH ON STOCHASTIC PERT NETWORKS:1966 - 1987. *INFOR*, 27(3), August:272-296.

Albrecht, A.J. & Gaffney, J.E. 1983. Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation. *IEEE Transactions on Software Engineering*, SE-9(6), November:639-648.

Allen, D. 1991. Management of quality. *Management Accounting*, 69(9), October:19.

Anderson, O. 1990. Use of software engineering data in support of project management. *Software Engineering Journal*, 5, November:350-356.

Andreou, S.A. 1991. Capital resource allocation for strategic quality management. *Int. J. Technology Management, Special Issue on Manufacturing Strategy*, 6(3/4): 415-426.

Apics Dictionary (The Official Dictionary of Production and Inventory Management Terminology and Phrases). 1987.S.v. "JIT". Sixth Edition. Falls Church, VA: The American Production and Inventory Control Society.

Archibald, R.D. & Villoria, R.L. 1967. *Network-Based Management Systems (PERT/CPM)* New York: Wiley.

Arifoglu, A. 1993. A methodology for software cost estimation. *ACM Sigsoft, Software Engineering Notes,* 18(2), April: 96-105.

Arnold, K.L. 1994. *The manager's guide to ISO 9000.* New York: THE FREE PRESS.

Babu, A.J.G. & Suresh, N. 1996. Project management with time,cost, and quality considerations.*European Journal of Operational Research,* 88(2): 320-327.

Banker, R.D., Chang, H. & Kemerer, C.F. 1994. Evidence on economies of scale in softwaredevelopment. *Information and Software Technology,* 36(5): 275-282.

Banker, R.D. & Kemerer, C.F. 1989. Scale Economies in New Software Development. *IEEE Transactions on Software Engineering,* SE-15(10), October: 1199-1205.

Barrier, M. 1992. Small Firms Put Quality First. *Nation's Business,* May:22-32.

Basili, V.R. & Musa, J.D. 1991. The Future Engineering of Software: A Management Perspective.*Computer,* 24(9), September: 90-96.

Basili, V.R. & Weiss, D.M. 1984. A Methodology for Collecting Valid Software Engineering Data. *IEEE Transactions on Software Engineering,* SE-10(6), November: 728-738.

Bazzana,G., Zontini,G., Damele,G., Maiocchi,M. & Giunchi,M. 1993. Applying software reliability models to a large industrial dataset. *Information and Software Technology,* 35 (11/12), November/December: 669-677.

Bendell, A., Solomon, D. & Carter, J.M. 1995. Evaluating Project Completion Times when Activity Times are Erlang Distributed *Journal of the Operational Research Society,* 46(7), July: 867-882.

Benson, P.G., Saraph, J.V. & Schroeder, R.G. 1991. The Effects of Organizational Context on Quality Management: An Empirical Investigation *Management Science*, 37(9), September: 1107-1124.

Benson, T. 1992. IQS^{SM}: Quality is not what you think it is. *Industry Week*, 5 October: 22-34.

Berny, J. 1989. A New Distribution Function for Risk Analysis. *Journal of the Operational Research Society*, 40(12): 1121-1127.

Berry, L.L., Parasuraman, A. & Zeithaml, V.A. 1988. The Service-Quality Puzzle. *Business Horizons*, 31(5): 35-43.

Boehm, B.W. 1981. *Software engineering economics.* Englewood Cliffs, NJ: Prentice-Hall.

Bollinger,T.B. & McGowan, C.A. 1991. A Critical Look at Software Capability Evaluations. *IEEE Software*, July: 25-41.

Bonett, D.G. & Deckro, R.F. 1993. A multinomial project evaluation and review technique for information systems analysis and design. *Information and Management*, 25: 51-55.

Booch, G. 1991. *Object-Oriented Design: with applications.* Redwood City, Calif.: Benjamin/Cummings.

BOOTSTRAP: EUROPE'S ASSESSMENT METHOD. 1993. *IEEE Software,* May: 93-95.

Bossert, J.L. 1991. *Quality Function Deployment: A practioner's approach.* Milwaukee, Wis.: ASQC Quality Press.

Britney, R.R. 1976. Bayesian point estimation and the PERT scheduling of stochastic activities. *Management Science*, 22(9): 938-948.

Broeker, E.J. 1989. Build a Better Supplier-Customer Relationship. *Quality Progress*, 22(9), September: 67-68.

Brown, M.G. 1989. Commitment...It's not the Whether, It's the " How To". *Journal for Quality and Participation*, December: 38-42.

Brown, M.G. 1991. The Baldrige Award: How do you win? *Automation*, October: 34-37.

Capacci, C. & Stamelos, I. 1994. Constructing software cost models. *CSELT Technical Reports*, 22(1), February: 27-43.

Carpenter, G.D. 1991. Total quality management: A journey to environmental excellence. *Environment Today*, May: 27,45.

Carroll, L. Alice in Wonderland (1865), in *The Complete Illustrated Works of LEWIS CARROLL*. 1982. London: Chancellor Press.

Carter, G.D., Clare, C.P. & Thorogood, D.C.J. 1987. Engineering project management techniques and their application to computer projects. *Software Engineering Journal*, 2(1), January: 15-20.

Cassell, R.H. 1992. Seven Steps to a Successful Customer Survey. *Quality Progress*, 25(7), July: 65-68.

Chaleff, I. 1993. What quality-conscious companies are overlooking. *QUALITY*, 32(5), May: 49-51.

Chambers, J.M. & Hastie, T.J. 1992. *Statistical models in S.* Pacific Grove: Wadsworth & Brooks.

Chidamer, S.R. & Kemerer, C.F. 1991. Towards a metrics suite for object oriented design, in *OOPSLA '91*, New York: ACM Press: 197-211.

Clapp, J. 1993. Getting started on software metrics. *IEEE Software*, January: 108-109,117.

Clark, C.E. 1962. The PERT model for the distribution of an activity time. *Operations Research*, 10: 405-406.

Coad, P. & Yourdon, E. 1991. *Object-Oriented Design*. Englewood Cliffs, NJ: Yourdon Press.

Conte, S.D., Dunsmore, H.E. & Shen, V.Y. 1986. *Software Engineering Metrics and Models*. Menlo Park, Calif.: Benjamin/Cummings.

Cravens, D.W., Holland, C.W., Lamb, C.W. & Moncrief, W.C. 1988. Marketing's Role in Product and Service Quality. *Industrial Marketing Management*, 17, November: 285-304.

Crosby, P.B. 1979. *QUALITY IS FREE: The Art of Making Quality Certain.* New York: McGraw-Hill.

Crosby, P.B. 1992. The Next Effort. *Management Review*, 81(2): 64.

Currit, P.A., Dyer, M. & Mills, H.D. 1986. Certifying the Reliability of Software. *IEEE Transactions on Software Engineering*, SE-12(1), January: 3-11.

Cusumano, M.A. 1991. *Japan's Software Factories: A Challenge to U.S. Management.* New York: Oxford University Press.

Dale, B.G. & Plunkett, J.J. 1991. *Quality Costing*. London: Chapmann and Hall.

Daskalantonakis, M.K. 1992. A Practical View of Software Measurement and Implementation Experiences Within Motorola. *IEEE Transactions on Software Engineering*, 18(11), November: 998-1010.

Davis, T.R.V. 1992. Baldridge Winners LInk Quality, Strategy, and Financial Management. *Planning Review*, November/December: 36-40.

Deckro, R.F., Hebert, J.E., Verdini, W.A., Grimsrud, P.H. & Venkateshwar, S. 1995. Nonlinear time/cost tradeoff models in project management. *Computers and Industrial Engineering,* 28(2): 219-229.

DeMarco, T. 1982. *Controlling software projects: Management, Measurement & Estimation.* New York: Yourdon Press

Donaldson, W. 1965. Estimation of the mean and variance of a PERT activity time. *Operations Research,* 13: 382-385.

Draper, N.R. & Smith, H. 1966. *Applied Regression Analysis.* New York: Wiley.

Dyer, M. 1992. *The Cleanroom Approach to Quality Software Development.* New York: Wiley.

Edwards, J.S. & Moores, T.T. 1994. A conflict between the use of estimating and planning tools in the management of information systems. *European Journal of Information Systems,* 3(2): 139-147.

Elmaghraby, S.E. 1977. *ACTIVITY NETWORKS: Project Planning and Control by Network Models.* New York: Wiley.

Elmaghraby, S.E. 1995. Activity nets: A guided tour through some recent developments. *European Journal of Operational Research,* 82: 383-408.

Erikkson, I. & McFadden, F. 1993. Quality function deployment: a tool to improve software quality. *Information and Software Technology,* 35(9), September: 491-498.

Esterling, B. 1980. Software Manpower Costs: A Model. *Datamation,* March: 164-170.

Farnum, N.R. & Stanton, L.W. 1987. Some results concerning the estimation of Beta distribution parameters in PERT. *Journal of the Operational Research Society,* 38(3): 287-290.

Fechter, W.F. 1993. The Competitive Myth. *Quality Progress*, 26(5), May: 87-90.

Fenton, N.E. 1991. *SOFTWARE METRICS: A rigorous approach*. 1st edition. London: Chapman & Hall.

Fenton, N. & Whitty, R. 1995. Introduction, in *Software Quality Assurance and Measurement: A Worldwide Perspective*, editors N. Fenton, R. Whitty, Y.Iizuka. London: International Thomson Computer Press: 1-19.

Fenwick, A.C. 1991. Five Easy Lessons: A primer for starting a TQM program. *Quality Progress*, 24(12), December: 63-66.

Finley, E.D. & Fisher, D.J. 1994. Project scheduling risk assessment using Monte Carlo methods. *Cost Engineering*, 36(10), October: 24-28.

Flynn, B.B., Schroeder, R.G. & Sakakibara, S. 1994. A framework for quality management research and an associated measurement instrument. *Journal of Operations Management*, 11(4), March: 339-366.

Forker, L.B. 1991. Quality: American, Japanese and Soviet perspectives. *Academy of Management Executive*, 5(4), November: 63-74.

Fox, C., Levitin, A. & Redman, T. 1994. The notion of data and its quality dimensions. *Information Processing and Management*, 30(1): 9-19.

Gallagher, C. 1987. Reply: A note on PERT assumptions. *Management Science*, 33(10), October: 1360.

Garvin, D.A. 1983. Quality on the line. *Harvard Business Review*, September-October: 65-75.

Garvin, D.A. 1984. What Does "Product Quality" Really Mean? *Sloan Management Review*, Fall: 25-43.

Garvin, D.A. 1988. *Managing Quality: the strategic and competitive edge*. New York: Free Press.

Godfrey, A.B. 1993. Ten Areas for Future Research in Total Quality Management. *Quality Management Journal*, 1(1), October: 47-70.

Golenko-Ginzburg, D. 1989. A New approach to the Activity-time Distribution in PERT. *Journal of the Operational Research Society*, 40(4): 389-393.

Goodman, P.A. 1992. Application of cost estimation techniques: Industrial perspective. *Information and Software Technology*, 34(6), June: 379-382.

Gordon, P.J. 1993. Customer satisfaction research reaps rewards. *QUALITY*, 32(5), May: 39-41.

Grady, R. 1992. *Practical Software Metrics for Project Management and Process Improvement*. Englewood Cliffs, NJ: Prentice-Hall.

Grady, R. 1994. Successfully Applying Software Metrics. *COMPUTER*, September: 18-25.

Grady, R.B. & Caswell, D. 1987. *Software Metrics: Establishing a Company-Wide Program*. Englewood Cliffs, NJ: Prentice-Hall.

Grubbs, F. 1962. Attempts to validate certain PERT statistics or "picking on PERT". *Operations Research*, 10: 912-915.

Harari, O. 1993. Ten Reasons Why TQM Doesn't Work. *Management Review*, 82(1), January: 33-38.

Hastings, N.A.J. & Peacock, J.B. 1974. *Statistical Distributions: A handbook for students and practioners*. London: Butterworths.

Hauser, J.R. & Clausing, J. 1988. The House of Quality. *Harvard Business Review*, May-Jun: 63-73.

Hayes, R.H. 1981. Why Japanese Factories Work. *Harvard Business Review*, July-August: 57-66.

Heemstra, F.J. 1992. Software cost estimation. *Information and Software Technology*, 34(10), October: 627-639.

Hetzel, B. 1993. *Making Software Measurement Work: Building an Effective Measurement Program*. Boston: QED Publishing Group.

Hill, D. 1992. These Foolish Things. *Quality Progress*, 25(7), July: 73-74.

Holcomb, M.C. 1994. Customer Service Measurement: A methodology for increasing customer value through utilization of the Taguchi strategy. *Journal of Business Logistics*, 15(1): 29-52.

Horst, R.L. 1992. Investing in Automation: Total Quality Management Unlocks the Dollars. *Control & Systems*, January: 46-48.

Huda, F. & Preston, D. 1992. Kaizen: the applicability of Japanese techniques to IT. *Software Quality Journal*, 1: 9-26.

Hughes, R.G. 1985. The safety management maturity grid. *Professional Safety*, 30(6), June: 15-18.

Humphrey, W.S. 1988. Characterizing the Software Process: A Maturity Framework. *IEEE Software*, March: 73-79.

Humphrey, W.S. 1989. *Managing the Software Process*. Reading, Mass.: Addison-Wesley.

Humphrey, W.S. 1991. Recent findings in software process maturity, in *Software Development Environments and CASE Technology. European Symposium Proceedings.* Berlin, Germany: Springer-Verlag: 258-270.

Hyde, A.C. 1990-91. Rescuing Quality Measurement from TQM. *Bureaucrat,* 19(4), Winter: 16-20.

Inman, R.A. 1990. Quality Certification of Suppliers by JIT manufacturing firms. *Production and Inventory Management Journal,* 31(2): 58-61.

I/S ANALYZER: Special Issue on Software Engineering: Productivity and Quality. 1994. 32(2): 1-18.

Ishikawa, K. 1990. *Introduction to quality control.* London: Chapman and Hall.

Jacobs, D.A., Luke, S.R. & Reed, B.M. 1995. Using quality function deployment as a framework for process measurement. *Engineering Management Journal,* 7(2), June: 5-9.

Jeffery, R. & Berry, M. 1993. A Framework for Evaluation and Prediction of Metrics Program Success, in *Proceedings First International Software Metrics Symposium.* Los Alamitos, CA: IEEE Comput. Soc. Press: 28-39.

Jones, C. 1986. *Programming Productivity.* New York: McGraw-Hill.

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

Jones, C. 1993. *Software Productivity and Quality Today: The Worldwide Perspective.* Carlsbad, CA: IS Management Group.

Jordan, J. 1992-1993. Everything you wanted to know about TQM. *Public Manager,* 21(4), Winter: 45-48.

Kamburowski, J. 1985. Normally Distributed Activity Durations in PERT Networks. *Journal of the Operational Research Society*, 36(11): 1051-1057.

Kanter, R.M. 1987. Quality Leadership and Change. *Quality Progress,* February: 45-51.

Kaplan, R.S. & Norton, D.P. 1992. The Balanced Scorecard - Measures that drive performance. *Harvard Business Review*, January-February: 71-79.

Karlöf, B. & Östblom, S. 1993. *Benchmarking: A Signpost to Excellence in Quality and Productivity*. Chichester: Wiley.

Karnes, C.L., Sridharan, S.V. & Kanet, J.J. 1995. Measuring quality from the consumer's perspective: A methodology and its application. *International Journal of Production Economics*, 39(3): 215-225.

Keefer, D.L. 1994. Certainty Equivalents for Three-point Discrete-distribution Approximations. *Management Science*, 40(6), June: 760-773.

Keefer, D.L. & Bodily, S.E. 1983. Three-point approximations for continuous random variables. *Management Science,* 29(5), May: 595-609.

Keefer, D.L. & Verdini, W.A. 1993. Better Estimation of PERT Activity Time Parameters. *Management Science*, 39(9), September: 1086-1091.

Keith, R.B.(Jr.) 1994. MIS + TQM = QIS. *Quality Progress,* 27(4), April: 29-32.

Kemerer, C.F. 1987. An Empirical Validation of Software Cost Estimation Models. *Communications of the ACM,* 30(5), May: 416-429.

Kemerer, C.F. & Porter, B.S. 1992. Improving the Reliability of Function Point Measurement: An Empirical Study. *IEEE Transactions on Software Engineering,* 18(11), November: 1011-1024.

Kendric, J. 1993. Study looks at TQM: Is it forging ahead or falling behind? *QUALITY*, 32(5), May: 13.

Kidd, J.B. 1987. A comparison between the VERT program and other methods of project duration estimation. *OMEGA International Journal of Management Science*, 15(2): 129-34.

Kim, M.W. & Liao, W.M. 1994. Estimating Hidden Quality Costs with Quality Loss Functions. *Accounting Horizons*, 8(1), March: 8-18.

Kitchenham, B.A. 1992. Empirical studies of assumptions that underlie software cost estimation models. *Information and Software Technology*, 34(4), April: 211-217.

Kitchenham, B.A. & Mellor, P. 1991. Data collection and analysis, in *Software Metrics: A rigorous approach*, by N.E. Fenton. 1st edition. London: Chapman & Hall.

Kitchenham, B.A. & Taylor, N.R. 1985. Software Project Development Cost Estimation. *Journal of Systems and Software*, 5(4): 267-278.

Klaber, L.G. 1993. Principles and Profits. *Executive Excellence*, 10(10), Oct:14.

Knotts, U.S. (Jr.), Parrish, L.G. (Jr.) & Evans, C.R. 1993. What does the U.S. Business Community really think about the Baldrige Award? *Quality Progress*, 26(5), May: 49-53.

Koch, G.R. 1993. Process assessment: the "BOOTSTRAP" approach. *Information and Software Technology*, 35(6/7), June/July: 387-403.

Laranjeira, L.A. 1990. Software Size Estimation of Object-Oriented Systems. *IEEE Transactions on Software Engineering*, 16(5), May: 510-522.

Lau, Hon-Shiang & Somarajan, C. 1995. A proposal on improved procedures for estimating task-time distributions in PERT. *European Journal of Operational Research*, 85(1), August: 39-52.

Lederer, A.L. & Prasad, J. 1992. Nine Management Guidelines for Better Cost Estimating. *Communications of the ACM*, 35(2), February: 51-59.

Lee, Kuen-Ching, Lu, Iuan-Yuan & Lin, Hsin-Hui. 1994. PM-Net: a software project management representation model. *Information and Software Technology*, 36(5): 295-308.

Lee, Gang-Soo. & Murata,T. 1994. A B-distributed Stochastic Petri Net Model for Software Project Time/Cost Management. *Journal of Systems and Software*, 26: 149-165.

Lee, D.E. & Willis. V.J. 1988. Mining for Gold (Quality) in Trust Operations. *Trusts & Estates*, 127(3), March: 57-63.

Leonard, F.S. & Sasser, W.L. 1982. The incline of quality. *Harvard Business Review*, September-October: 163-171.

Littlefield, T.K. & Randolph, P.H. 1987. Reply: An answer to Sasieni's question on PERT times. *Management Science*, 33(10), October: 1357-1359.

Littlechild, S.C. & Shutler, M.F. (eds). 1991. *Operations Research in Management*. New York: Prentice Hall.

Liu, Lung-Chun. & Horowitz, E. 1989. A Formal Model for Software Project Management. *IEEE Transactions on Software Engineering*, 15(10), October: 1280-1293.

Lorenz, M. & Kidd, J. 1994. *Object-Oriented Software Metrics: A Practical Guide.* Englewood Cliffs, NJ: Prentice Hall.

Lytz, R. 1995. Software metrics for the Boeing 777: a case study. *Software Quality Journal*, 4: 1-13.

M-BASE Data Collection and Storage System. User Manual. Version 3. 1993. Manchester: National Computing Centre.

Maani, K.E. 1988. Quality, Cost, Productivity and Profitability - Are they really compatible?, in *Asian-Pacific Operations Research: APORS '88*, edited by B.-H. Ahn. Amsterdam: Elsevier Science: 441-450.

MacCrimmon, K.R. & Ryavec, C.A. 1964. An Analytical Study of the PERT assumptions. *Operations Research*, 12: 16-37.

MacDonell, S.G. 1994. Comparative review of functional complexity assessment methods for effort estimation. *Software Engineering Journal*, 9(3), May: 107-116.

Malcolm, D.G., Roseboom, J.H., Clark, C.E. & Fazar,W. 1959. Applications of a technique for Research and Development Program Evaluation. *Operations Research*, 7: 646-669.

Marshall, I.M., Samson, W.B., Dugard, P.I. & Scott, W.A. 1994. Predicting the development effort of multimedia courseware. *Information and Software Technology*, 36(5): 251-258.

Matson, J.E., Barrett, B.E. & Mellichamp, J.M. 1994. Software Development Cost Estimation Using Function Points. *IEEE Transactions on Software Engineering*, 20(4), April: 275-287.

Merkhofer, M.W. 1987. Quantifying Judgmental Uncertainty: Methodology, Experiences, and Insights. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-17(5), September/October: 741-752.

MERMAID MARK 1P USER GUIDE - VERSION 1. 1992.

Miller, C. 1992. TQM's value criticized in new report. *Marketing News*, 26(23), November:1,16.

Mills, H.D., Dyer, M. & Linger, R.C. 1987. Cleanroom Software Engineering. *IEEE Software*, September: 19-25.

MIMS KEY PERFORMANCE INDICATORS. 1994. Brisbane: MINCOM.

Miyazaki, Y., Takanou, A., Nozaki, H., Nakagawa, N. & Okada, K. 1991. Method to estimate parameter values in software prediction models. *Information and Software Technology*, 33(3), April: 239-243.

Moder, J.J. & Rodgers, G.G. 1968. Judgment estimates of the moments of PERT type distributions. *Management Science*, 15(2), October:B76 - B83.

Moller, K.H. & Paulish, D.J. 1993. *SOFTWARE METRICS: A Practioner's guide to improved product development.* London: Chapman & Hall.

Mon, Don-Lin., Cheng, Ching-Hsue. & Lu, Han-Chung. 1995. Applications of fuzzy distributions on project management. *Fuzzy Sets and Systems*, 73: 227-234.

Monoky, J.F. 1992. Unleash the power of TQM. *Industrial Distribution*, 81(8), 15 June: 77.

Montana, A.J. 1992. If it isn't perfect, make it better. *Research Technology Management*, 35(4), July-August: 38-41.

Moores, T.T. & Edwards, J.S. 1992. Could large UK corporations and computing companies use software cost estimating tools? - a survey. *European Journal of Information Systems*, 1(5): 311-319.

Moroney, M. 1950. *Facts from Figures*. Harmonworth, Middlesex: Penguin Books.

Mowen, J.C., Licata, J.W. & McPhail, J. 1993. Waiting in the Emergency Room: How to improve Patient Satisfaction. *Journal of Health Care Marketing*, 13(2), Summer: 26-33.

Musa, J.D., Iannino, A. & Okumoto, K. 1990. *Software Reliability: Measurement, Prediction, Application.* Professional Edition. New York: McGraw-Hill.

Nakhai, B. & Neves, J.S. 1994. The Deming, Baldrige, and European Quality Awards. *Quality Progress*, 27(4), April: 33-38.

NCC FACT SHEET: An Introduction to Software Metrics. 1991. *PC Business Software,* 16(4): 6-9.

Oxford Advance Learner's Dictionary of Current English. 1974. S.v. "perspective". 3rd edition. Oxford: Oxford University Press.

Oxford Universal Dictionary. 1981. S.v. "consumerism", "process". Oxford: Oxford University Press.

Parasuraman, A., Zeithaml, V.A. & Berry, L.L. 1985. A Conceptual Model of Service Quality and Its Implications for Future Research. *Journal of Marketing*, 49, Fall:41-50.

Parasuraman, A., Zeithaml, V.A. & Berry, L.L. 1988. SERVQUAL: A Multiple-Item Scale for Measuring Customer Perceptions of Service Quality. *Journal of Retailing*, 64(1), Spring:12-40.

Parks, W.H. & Ramsing, K.D. 1965. The use of the Compound Poisson in PERT. *Management Science*, 15(8):B397-B402.

Paulk, M.C., Curtis, B., Chrissis, M.B. & Weber, C.V. 1993. Capability Maturity Model, Version 1.1. *IEEE Software,* July:18-27.

Pearson, E.S. & Tukey, J.W. 1965. Approximate means and standard deviations based on distances between percentage points of frequency curves. *Biometrika*, 52:533-546.

Perry, C. & Greig, I.D. 1975. Estimating the mean and variance of subjective distributions in PERT and Decision Analysis. *Management Science,* 21(12), August:1477-1480.

Perry, J.H. 1987. A Conceptual Framework for Evaluating Capital Equipment Purchases. *Journal of Purchasing and Materials Management*, Winter:32-38.

Perry, J.H. 1992. Evaluating Computer-Integrated Manufacturing Systems: The Role of Quality. *Production and Inventory Management Journal*, 33(2), Second Quarter:18-23.

Pfleeger, S.L. 1991a. Model of software effort and productivity. *Information and Software Technology*, 33(3), April:224-231.

Pfleeger, S.L. 1991b. Process maturity as framework for CASE tool selection. *Information and Software Technology*, 33(9), November:611-615.

Pfleeger, S.L. 1992. Measuring software reliability. *IEEE Spectrum*, 29(8), August: 56-60.

Pfleeger, S.L. 1993. Lessons learned in building a corporate metrics program. *IEEE SOFTWARE*, May: 67-74.

Pfleeger, S.L. & Fitzgerald, J.C.(Jr.). 1991. Software metrics tool kit: support for selection, collection and analysis. *Information and Software Technology*, 33(7), September:477-482.

Pfleeger, S.L. & McGowan, C. 1990. Software Metrics in the Process Maturity Framework. *Journal of Systems and Software*, 12: 255-261.

Pittman, M. 1993. Lessons Learned in Managing Object-Oriented Development. *IEEE SOFTWARE*, January: 43-51.

Pollack-Johnson, B. 1995. Hybrid structures and improving forecasting and scheduling in project management. *Journal of Operations Management*, 12(2), February: 101-117.

Pressman, R.S. 1993. *A Manager's Guide to Software Engineering*. New York: McGraw-Hill.

Pritsker, A.A.B. & Happ, W.W. 1966. GERT: Graphical Evaluation and Review Technique: Part 1. Fundamentals. *Journal of Industrial Engineering,* 17(5): 267-274.

Pritsker, A.A.B. & Whitehouse, G.E. 1966. GERT: Graphical Evaluation and Review Technique: Part II. Probabilistic and Industrial Engineering Applications. *Journal of Industrial Engineering,* 17(6):293-301.

Putnam, L.H. & Fitzsimmons, A. 1979. *Estimating Software Costs*. Datamation, September: 189-198.

Quigley, C. & McNamara, C. 1992. Evaluating Product Quality: An Application of the Taguchi Quality Loss Concept. *International Journal of Purchasing and Materials Management*, 28(3), Summer:19-25.

Ranasinghe, M. 1994. Quantification and management of uncertainty in activity duration networks. *Construction Management and Economics*, 12(1), January:15-29.

Rugg, D. 1993. Using a Capability Evaluation to Select a Contractor. *IEEE Software*, July:36-45.

Ryś,T., Stanek, R. & Ziembla, W. 1994. MIPS: A DSS for multiobjective interactive project scheduling. *European Journal of Operational Research*, 79:196-207.

S/PLUS for Windows User Manual. Vol. 2. 1993. Seattle, Washington: STATSCI.

S/PLUS for Windows Version 3.2. Supplement. 1994. Seattle, Washington: STATSCI.

Saaty, T.L. 1980. *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation.* New York: McGraw-Hill.

Safford, R.R., Gobeli, D.H. & Suen,K.P. 1990. OMAX+ : An advanced quality improvement measurement system. *Computers and Industrial Engineering*, 19:267-271.

Salter, J.M. (II) 1991. Take systematic approach to measuring satisfaction. *Marketing News*, 25(3), 4 Feb: 9.

Saraph, J.V., Benson, P.G. & Schroeder, R.G. 1989. An instrument for Measuring the Critical Factors of Quality Management. *Decision Sciences*, 20:810-829.

Sasieni, M.W. 1986. A note on PERT times. *Management Science*, 32:1652-1653.

Schaffer, R.H. & Thomson, H.A. 1992. Successful Change Programs Begin with Results. *Harvard Business Review*, January-February:80-89.

Schrock,E.M. & Lefevre,H.L. 1988. *The good and the bad news about quality*. New York: Marcel Dekker.

Schmitthenner, J.W. (III) 1993. METRICS. *Management Accounting*, May:27-30.

Schulmeyer, G.G. and McManus, J.I. 1992. *Total Quality Management for Software*. New York: Van Nostrand Reinhold.

Sculli, D. 1983. The Completion Time of PERT networks. *Journal of the Operational Research Society*, 34(2):155-158.

Selvidge, J. 1980. Assessing the extremes of probability distributions by the fractile method. *Decision Sciences*, 11:493-502.

Sheldon, F.T., Kavi, K.M., Tausworthe, R.C., Yu,J.T., Brettschneider,R. & Everett, W.W. 1992. Reliability Measurement: From Theory to Practise. *IEEE Software*, July:13-20.

Shin, Sueng-Il., Riel, P.F. & Sink, D.S. 1988. Using Decision Tree and Contour Map Approaches for the selection of the Performance Measurement Techniques, in *Asian-Pacific Operations Research: APORS '88*, editor B.-H. Ahn. Amsterdam: Elsevier Science:451-460.

Sichel, H.S. 1975. On a distribution law for word frequencies. *Journal of the American Statistical Association*, 70(351), September:542-547.

Smith, G.F. 1993. The meaning of quality. *TOTAL QUALITY MANAGEMENT*, 4(3):235-244.

Spendolini, M.J. 1992. *The Benchmarking Book*. New York: AMACOM.

Splendolini, M.J. 1993. How to buikd a benckmarking team. *Journal of Business Strategy,:* 53-57.

Stanleigh, M. 1992. Accounting for Quality. *CA Magazine*, 125(10), October:40-42.

Stark,G. & Durst, R.C. 1994. Using Metrics in Management Decision Making. *COMPUTER*, September:42-48.

Stevenson,C. 1995. *Software Engineering Productivity. A practical guide.* London: Chapmann & Hall.

Stout, G. 1993. Customers drive CQI in Telecommunications. *QUALITY*, 32(5), May:18-26.

Sweet, E.B. (Jr.) Put Purchasing on the Grid. *Purchasing World,* 27(9), September:69-70.

Symons, R.C. 1988. Function Point Analysis: Difficulties and Improvements. *IEEE Transactions on Software Engineering*, 14(1), January:2-11.

Thomas, R., Fernstroem, C. & Hesse, F. 1991. ESF - A European Programme for Evolutionary Introduction of Software Factories. *Software World*, 22(1):10-14.

Thornton, D. 1994. A quality assurance reference model for object-orientation. MSc (Information Systems) dissertation, UNISA, Pretoria.

Urban, J.J. 1993. Harnessing human resource technologies. *Business Quarterly*, 58(1), Autumn:71-75.

Van der Merwe, C.A. 1989. Total Quality - What does it involve? Conference on Quality and Reliability, May 1989, Johannesburg.

Van Genuchten, M. & Koolen,H. 1991. On the use of software cost estimation models. *Information and Management*, 21: 37-44.

Van Slyke, R.M. 1963. Monte Carlo Methods and the PERT problem. *Operations Research*, 11, September: 839-860.

Venables, W.N. & Ripley, B.D. 1994. *Modern Applied Statistics with S/PLUS*. New York: Springer-Verlag.

Verner, J. & Tate, G. A Software Size Model. *IEEE Transactions on Software Engineering*, 18(4), April: 265-278.

Wagner, H.M. 1984. Profit Wonders, Investment Blunders. *Harvard Business Review*, September-October: 121-135.

Ware, R. 1994. Does technology or technique deliver projects? *Journal of Systems Management*, 45, July: 10.

Welch, J.F. 1992. Service Quality Measurement at American Express Traveler's Cheque Group. *National Productivity Review*, 11(4), Autumn: 463-471.

Wellman, F. 1993. *Software Costing*. New York: Prentice Hall.

Westney, R. 1989. Getting Personal: The Missing Link - Found at Last? *Cost Engineering*, 31(4), April: 28-29.

Whiting, R. 1991. Benchmarking: Lessons from the best-in-class. *Electronic Business*, 17(19), October: 128-134.

Whitten, N. 1995. *Managing Software Development Projects. Formula for Success.* Second Edition, New York: Wiley.

Williams, T.M. 1992. Practical Use of Distributions in Network Analysis. *Journal of the Operational Research Society*, 43(3):265-270.

Wood, M. & Preece, D. 1992. Using Quality Measurements: Practise, Problems and Possibilities. *International Journal of Quality and Reliability Management*, 9(7): 42-53.

Yovovich, B.G. 1991. Becoming a World-Class Customer. *Business Marketing (IMR)*, 76(9), September: 16, 29.

Zeitmahl, V.A. 1987. *Defining and Relating Price, Perceived Quality, and Perceived Value.* Cambridge, MA: Marketing Science Institute, Report No 87-101..

APPENDIX A

THE INITIAL 78 ITEMS USED FOR MEASURING THE CRITICAL FACTORS OF QUALITY MANAGEMENT

This appendix contains the items contained in the original research instrument. The items noted by an asterisk (*) were eventually dropped to improve the reliability of the instrument.

Factor 1: **Role of divisional top management and quality policy**

1 Extent to which the top division executive (responsible for division profit and loss) assumes responsibility for quality performance.

2 Acceptance of responsibility for quality by major department heads within the division.

3 Degree to which divisional top management (top divisional executive and major department heads) is evaluated for quality performance.

4 Extent to which the division top management supports long-term quality improvement process.

5 Degree of participation by major department heads in the quality improvement process.

6 Extent to which the divisional top management has objectives for quality performance.

7 **Specificity of quality goals within the division.**

8 Comprehensiveness of the goal-setting process for quality within the division.

9 Extent to which quality goals and policy are understood within the division.

10 Importance attached to quality by the divisional top management in relation to cost and schedule objectives.

11 Amount of review of quality issues in divisional top management meetings.

12 Degree to which the divisional top management considers quality improvement as a way to increase profits.

13 Degree of comprehensiveness of the quality plan within the division.

Factor 2: **Role of the quality department**

14 Visibility of the quality department.

15 Quality department's access to divisional top management.

16 Autonomy of the quality department.

17* Utilization of quality staff professionals as a consulting resource.

18 Amount of coordination between the quality department and other departments.

19 Effectiveness of the quality department in improving quality.

Factor 3: **Training**

20 Specific work-skills training (technical and vocational) given to hourly employees throughout the division.

21* Team building and group dynamics training for employees in the division.

22 Quality-related training given to hourly employees throughout the division.

23 Quality-related training given to managers and supervisors throughout the division.

24 Training in the "total quality concept" (i.e., philosophy of company-wide

responsibility for quality) throughout the division.

25* Training of employees to implement quality circle type program.

26 Training in the basic statistical techniques (such as histograms and control charts) in the division as a whole.

27 Training in advanced statistical techniques (such as design of experiments and regression analysis) in the division as a whole.

28 Commitment of the divisional top management to employee training.

29 Availability of resources for employee training in the division.

Factor 4: Product/service design

30 Thoroughness of new product/service design reviews before the product/service is produced and marketed.

31 Coordination among affected departments in the product/service development process.

32 Quality of new products/services emphasized in relation to cost or schedule objectives.

33* Extent of analysis of customer requirements in product/service development process.

34 Clarity of product/service specifications and procedures.

35 Extent to which implementation/producibility is considered in the product/service design process.

36* Extent to which sales and marketing people consider quality a saleable attribute.

37 Quality emphasis by sales, customer service, marketing, and PR personnel.

Factor 5: Supplier quality management (supplier of goods and/or services)

38 Extent to which suppliers are selected based on quality rather than price or schedule.

39 Thoroughness of the supplier rating system.

40 Reliance on reasonably few dependable suppliers.

41 Amount of education of supplier by division.

42 Technical assistance provided to the suppliers.

43 Involvement of the supplier in the product development process.

44 Extent to which longer term relationships are offered to suppliers.

45 Clarity of specifications provided to suppliers.

46* Responsibility assumed by purchasing department for the quality of incoming products/services.

47* Extent to which suppliers have programs to assure quality of their products/services.

Factor 6: Process management/operating procedures

48 Use of acceptance sampling to accept/reject lots or batches of work.

49* Use of statistical control charts to control processes.

50 Amount of preventative equipment maintenance.

51 Extent to which inspection, review, or checking of work is automated.

52 Amount of incoming inspection, review, or checking.

53 Amount of in-process inspection, review, or checking.

54 Amount of final inspection, review, or checking.

55* Importance of inspection, review, or checking of work.

56* Self-inspection of work by workers.

57 Stability of production schedule/work distribution.
58 Degree of automation of the process.
59 Extent to which process design is "fool-proof" and minimizes the chances of employee errors.
60 Clarity of work or process instructions given to employees.

Factor 7: Quality data and reporting
61 Availability of cost of quality data in the division.
62 Availability of quality data (error rates, defect rates, scrap, defects, etc.)
63 Timeliness of the quality data.
64* Extent of quality data collected by the service/support areas of the division.
65 Extent to which quality data (cost of quality, defects, errors, scrap, etc.) are used as tools to manage quality.
66 Extent to which quality data are available to hourly employees.
67 Extent to which quality data are available to managers and supervisors.
68 Extent to which quality data are used to evaluate supervisor and managerial performance.
69 Extent to which quality data, control charts, etc., are displayed at employee work stations.

Factor 8: Employee relations
70 Extent to which quality circle or employee involvement type programs are implemented in the division.
71 Effectiveness of quality circle or employee involvement type programs in the division.
72 Extent to which employees are held responsible for error-free output.
73 Amount of feedback provided to employees on their quality performance.
74 Degree of participation in quality decisions by hourly/nonsupervisory employees.
75 Extent to which quality awareness building among employees is ongoing.
76 Extent to which employees are recognized for superior quality performance.
77* Impact of labor union quality improvement.
78 Effectiveness of supervisors in solving problems/issues.

APPENDIX B

The 26 *Measurement Items for Organizational Quality Context*

Managerial Knowledge

1. Experience in quality
2. Participation in professional quality organizations
3. Familiarity with various quality programs (Zero Defects, TQC)
4. Expertise in quality concepts
5. Overall knowledge of quality

Corporate Support for Quality

6. Level of corporate goal setting in quality
7. Rewards for corporate management for quality performance
8. Corporate leadership for quality
9. Quality viewed by corporate management as strategic variable
10. Corporate quality emphasis throughout the organization
11. Corporate management's commitment to quality
12. Progressiveness and innovativeness of corporate management
13. Corporate sanctioned rewards for divisional management for quality performance
14. Resources made available by corporate management for quality improvement
15. Appropriateness of corporate systems (plants, equipment, systems) for quality improvement

Marketplace Environment

16. Degree of competition faced by the business unit
17. Barriers to entry in the industry
18. Quality demands of customers
19. Quality demands due to legal and regulatory requirements

Product/process Environment

20. Rate of change of product/process
21. Proportion of product/service purchased outside
22. Degree of manufacturing content (as opposed to service content)
23. Degree of batch type process (as opposed to flow type process)
24. Degree of product complexity

Past Quality Performance

25. Last three years' quality performance
26. Perceived customer satisfaction for last three years

APPENDIX C

HEWLETT
PACKARD

SOFTWARE DEVELOPMENT
METRICS FORM

Instructions

Fill out the general information on this page and the detailed information on the following three pages for the project when it achieves initial release. Use the back of this page to provide additional comments. Fold the package together so that the return address is showing and send via internal mail.

General information

Project Contact: ____________________ Division: _______________

Project Name: _______________________ Release ID (version): ___________

Manufacturing Release Date: _____________________

General Category of Software:
_____ Firmware
_____ Systems (including OPSYS, Data Comm, Compilers, etc.)
_____ Applications
_____ Other (specify) ____________

Was Prototyping used in developing this project?
_____ not used
_____ evolutionary (Prototype evolves into a product; Prototype code is used in final project)
_____ simulation (Prototype used for user feedback, feasibility, and human factors verification; Prototype cost is not used in final product)

Number of installations expected in the first year?
Internal _____ External _____

Release Information

May we publish the Project Name in the Software Metrics Data Base?
_____ yes _____ no

May we publish the name of the *Project Contact* in the Software Metrics Data Base?
_____ yes _____ no

____
Project Contact's signature

Revision Date: 2/1/86

HEWLETT
PACKARD

SOFTWARE DEVELOPMENT
METRICS FORM

PEOPLE/TIME/COST

Project Name: ______________________ Release ID:_____________

ACTIVITIES	ENG. PAYROLL MONTHS	CALENDER MONTHS
Investigate/Spec.		
Design		
Implement		
Test		

TOTALS		

% of overtime (or undertime) = ______%

Instructions

Fill out the appropriate row for each life cycle activity.

Indicate undertime with a minus sign.

At MR send to:

Metrics Administrator
Software Engineering Lab
Building 26U
3500 Deer Creek Rd.
Palo Alto, CA 94304

Revision Date: 2/1/86

People/Time/Cost Definitions

Engineering Payroll Months

The sum of calendar payroll months attributed to each project engineer, including people doing testing, adjusted to exclude extended vacations and extended leaves. This does not include time project managers spend on management tasks.

Overtime (or undertime)

Engineering time over/under the 40 hour engineering week averaged over the duration of a project. % over/under time can be used as a normalization factor for engineering payroll months. Indicate undertime with a minus sign.

Investigate/External Specification

All activities relating to the investigation and external specifications of the project. This includes evaluating and reviewing project requirements and writing external specifications (ES).

Design

All activities relating to the high and low level design of the project. This includes development of the design, design reviews, and writing of the internal specifications (IS).

Implement

All activities relating to the implementation of the project. This includes coding, code walkthroughs, unit (informal, private) testing and correcting defects.

Test

All activities relating to system (formal, public) testing. This includes writing test plans, writing test code, system and integration testing, and debugging defects found during test activities.

Calendar Months

Time elapsed in calender months between specific project checkpoints. The total calendar time must equal the sum of the calendar times for individual activities.

The checkpoint signalling the end of the investigate / external specification phase for calendar months is approval of the ES.

The checkpoint signalling the end of the design phase for calender months is the approval of the IS.

The checkpoint signalling the end of the implement phase for calendar months is the start of system (formal, public) testing.

The checkpoint signalling the end of the test phase for calendar months is manufacturing release (MR).

HEWLETT
PACKARD

SOFTWARE DEVELOPMENT
METRICS FORM

PRE-RELEASE DEFECTS

Project Name: ____________________ Release ID: ____________________

▨ Dotted areas are optional

ACTIVITIES	DEFECTS INTRODUCED	DEFECTS FOUND	DEFECTS CLOSED
Investigate/Spec.	▨		
Design	▨		
Implement	▨		
Test	▨		

TOTALS			

Instructions

At the end of each activity, fill in defects found and defects closed, and update defects introduced. If defects are not collected during a particular activity, leave it blank rather than enter zero. However, it is strongly recommended that accurate information be kept for all activities.

At MRE send to:

Metrics Administrator
Software Engineering Lab
Building 26U
3500 Deer Creek Rd.
Palo Alto, CA 94304

Revision Date: 2/1/86

Defect Definitions

Defect

A defect is a deviation from the product specification or an error in the specification if the error could have been detected and would have been corrected. If the error could not possibly have been detected, or it could have been detected and would not have been corrected, then it is an enhancement, not a defect. Defects do not include typographical or grammatical errors in the engineering documentation.

Defects introduced

The number of defects attributed to a flaw in the output of a particular activity which might not be found until a later activity. Do not include duplicates.

Defects Found

The number of defects found in a particular activity. Do not include duplicates.

Defects Closed

The number of defects corrected in a particular activity (Closed Service Requests, as defined by STARS, or Resolved Defects, as defined by DTS). Do not include duplicates.

Examples

Investigate / External Specification

Defects can be found in a formal review of engineering documents produced; e.g. ES, functional models, etc.

Design

Defects can be found during design inspections or through modelling.

Implement

Defects can be found during code inspections or unit (informal, private) tests.

Test

Defects can be found during system (formal, public) or integration testing.

HEWLETT
PACKARD

SOFTWARE DEVELOPMENT
METRICS FORM

DELIVERED SIZE

Project Name: ____________________ Release ID: ____________________

Language A : ____________________ Language B: ____________________

Line Counter (or other technique): ____

▨ Dotted areas are optional

	LANGUAGE A	LANGUAGE B
NCSS		
Comment Lines		
Blank Lines		

% of Recycled Code		

# of Procedures	▨	▨
Bytes of Object Code	▨	▨

# Lines in Engineering Documentation	▨
# Figures in Eng. Documentation	▨

Instructions

Use an automatic line counter. If no tool is available, estimate NCSS, comment lines, and blank lines of code (confidence level = ______________ %).

At MR send to:

Metrics Administrator
Software Engineering Lab
Building 26U
3500 Deer Creek Rd.
Palo Alto, CA94304

Revision Date: 2/1/86

Delivered Size Definitions

Delivered Size

Those lines of code which go into the product delivered to the customer.

NCSS

Non-Comment Source Statements which include compiler directives, data declarations, and executable code. Each physical line of code is counted once. Each include file is counted once. Print statements are lines of code.

Comment Lines

Lines containing only comments. A commented executable line is counted as executable code, not as a comment. Blank lines are not counted as comment lines.

Engineering Documentation

Documentation not included in the source code or in end-user documentation, such as user's manuals, administrative guides, or tutorials. Any documentation or messages in files that are not source files or end-user documentation are engineering documentation.

Examples of lines of engineering documentation are text lines in the ES, IS, test plans, etc. If estimating lines of documentation, use 54 lines per page.

A figure is a diagram or pictorial illustration or textual matter. Examples are data-flow diagrams, hierarchy charts, etc.

Recycled Code

Code incorporated into this product that was either used intact or highly leveraged from a different product or another part of this product.

At MR send to:

Metrics Administrator
Software Engineering Lab
Building 26U
3500 Deer Creek Road
Palo Alto, CA 94304

SOFTWARE COST ESTIMATION MODEL FORM

INSTRUCTIONS

Please supply the general information and the available detailed information on the following pages for each project. Definitions are given on a separate attached page. Additional comments regarding the project at hand that influenced the development time will be appreciated.

A GENERAL INFORMATION

PARTICULARS REQUIRED	DATA
PROJECT NAME	
PROJECT DESCRIPTION	
STARTING DATE OF PROJECT	
END DATE OF PROJECT	
ACTUAL TOTAL COST OF PROJECT	
ESTIMATE (AT BEGINNING OF PROJECT) OF TOTAL COST	
PROGRAMMING LANGUAGE(S) USED	
TEAM SIZE	

B DETAIL INFORMATION

PARTICULARS REQUIRED	DATA
CLASSIFY PROJECT AS ORGANIC, SEMIDETACHED OR EMBEDDED	
DELIVERED SIZE IN KDSI	
ACTUAL EFFORT (IN PERSON-MONTHS)	
ESTIMATED EFFORT AT BEGINNING OF PROJECT (IN PERSON-MONTHS)	

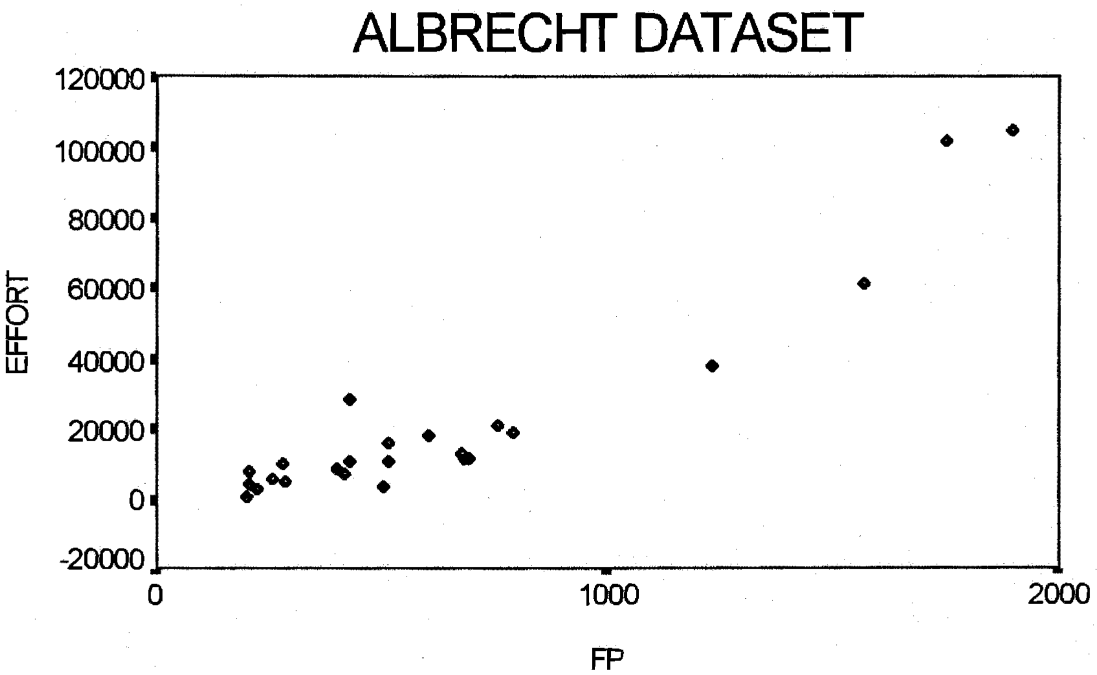
C ADDITIONAL INFORMATION

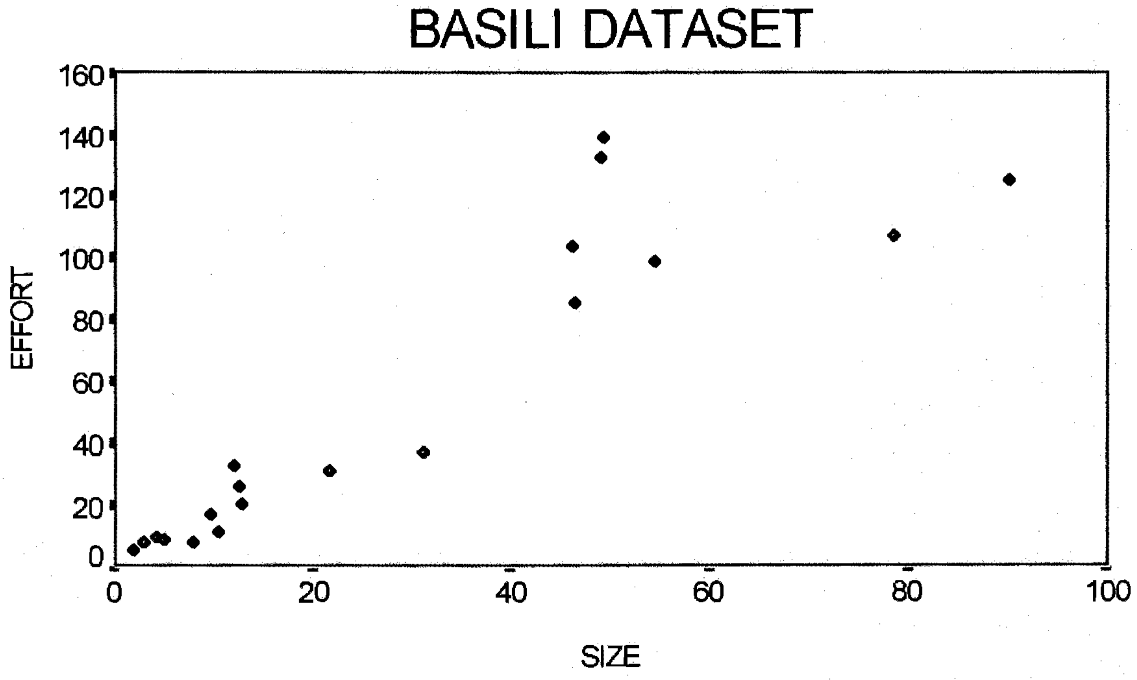
Please indicate the level of presence (high, average, low) of the following attributes:

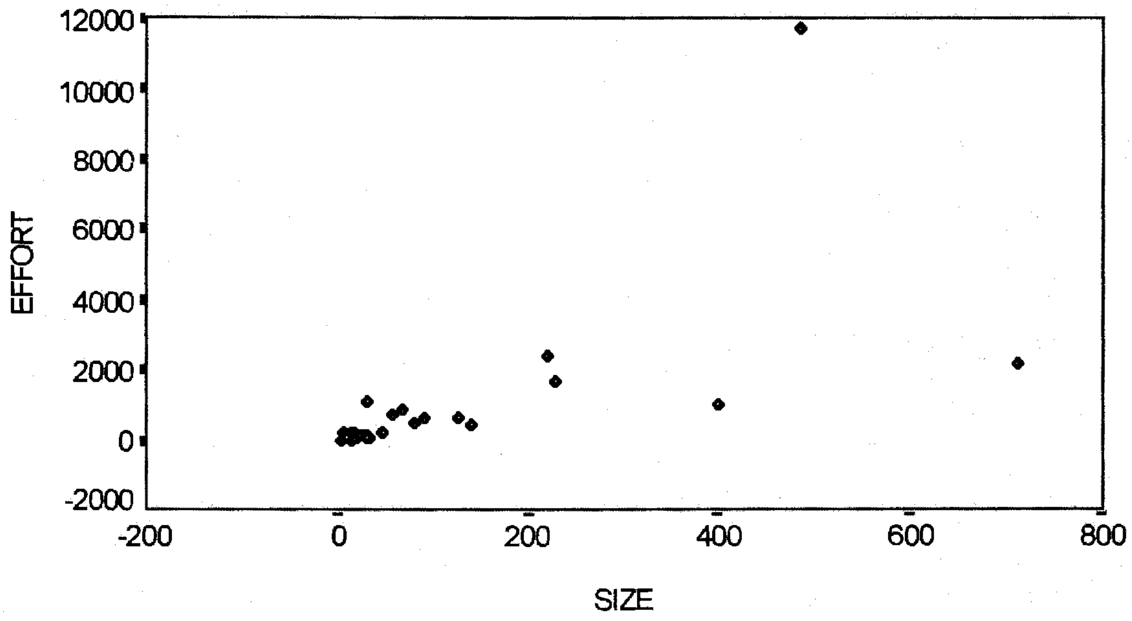
ATTRIBUTE	LEVEL (HIGH, AVERAGE, LOW)
USE OF SOFTWARE TOOLS	
SCHEDULE CONSTRAINTS	
PROGRAMMING LANGUAGE EXPERIENCE	
METHODOLOGY EXPERIENCE	
RESOURCE CONSTRAINTS	

D ADDITIONAL COMMENTS

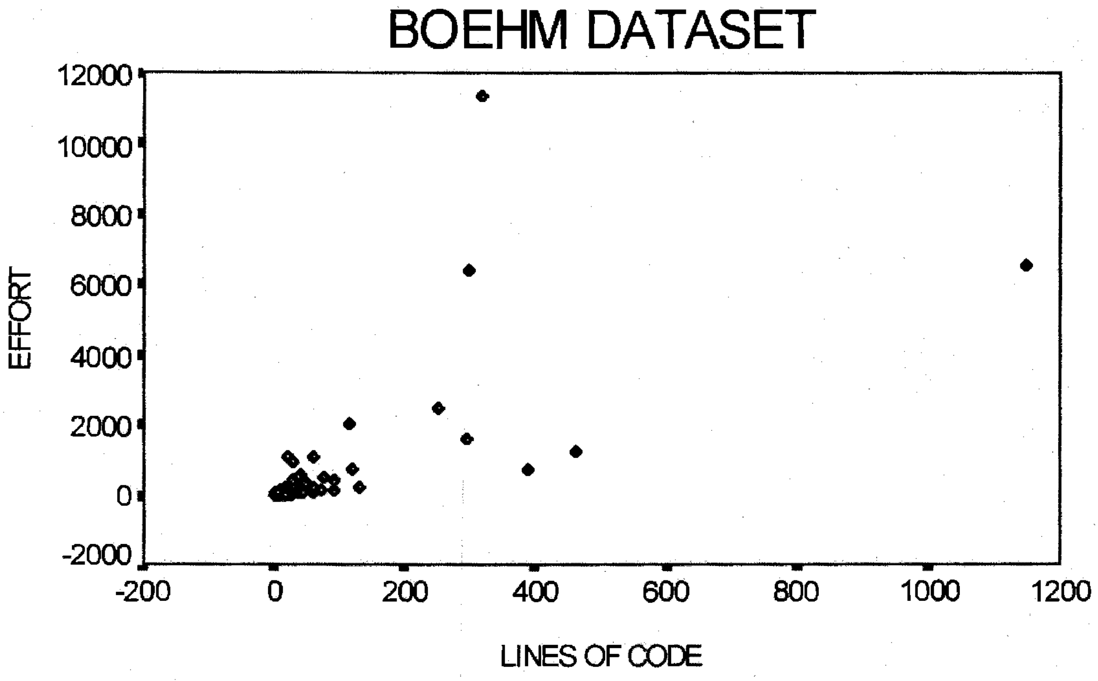
APPENDIX D

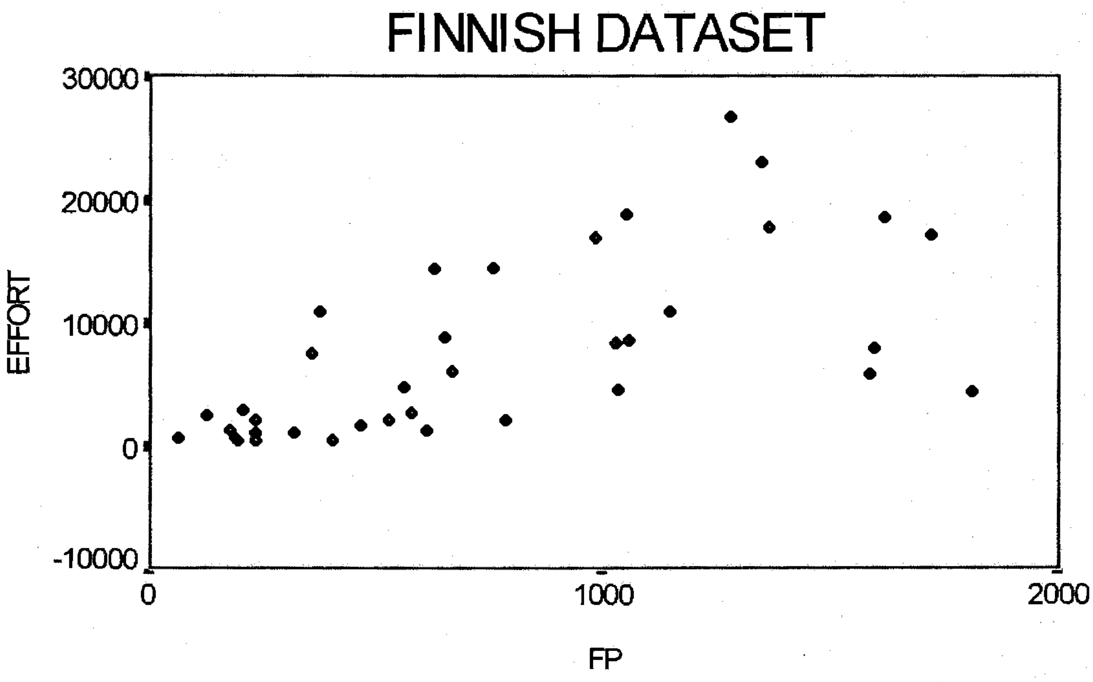

ALBRECHT DATASET
120000
100000
80000
60000
40000
20000
0
-20000
EFFORT
0
1000
2000
FP


BASILI DATASET
EFFORT
160
140
120
100
80
60
40
20
0
0
20
40
60
80
100
SIZE

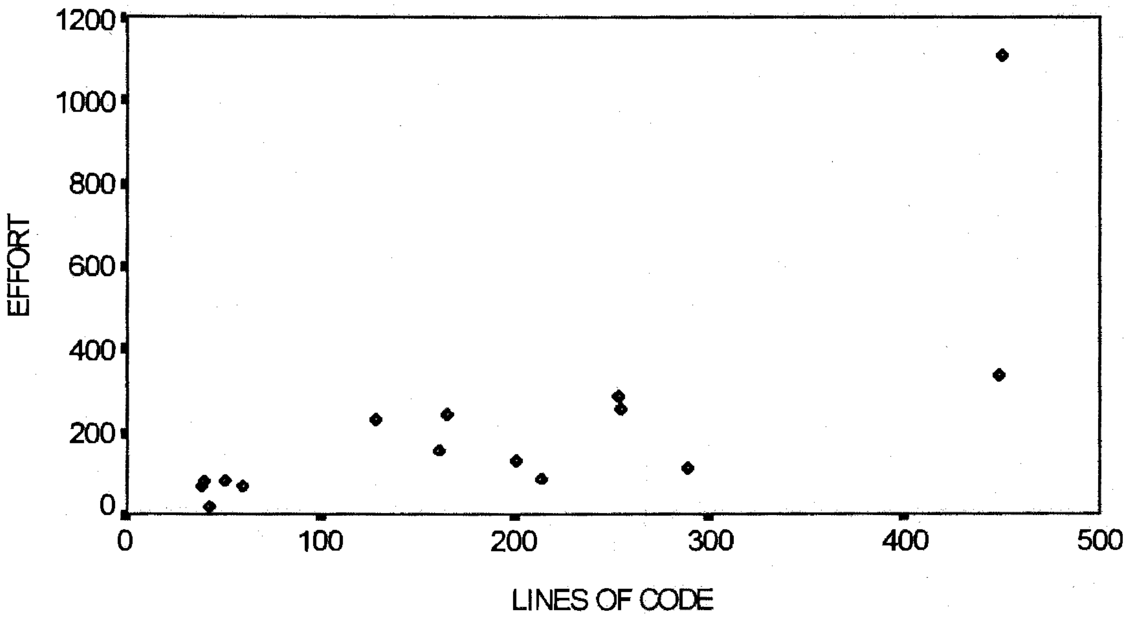


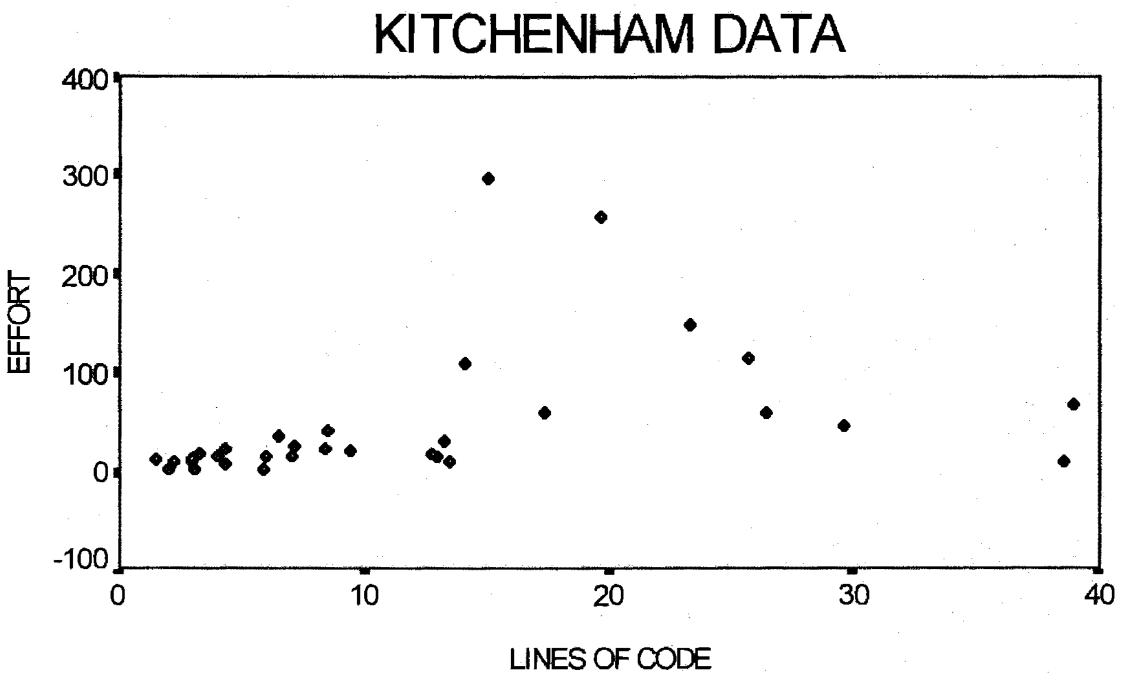
BELADY DATASET

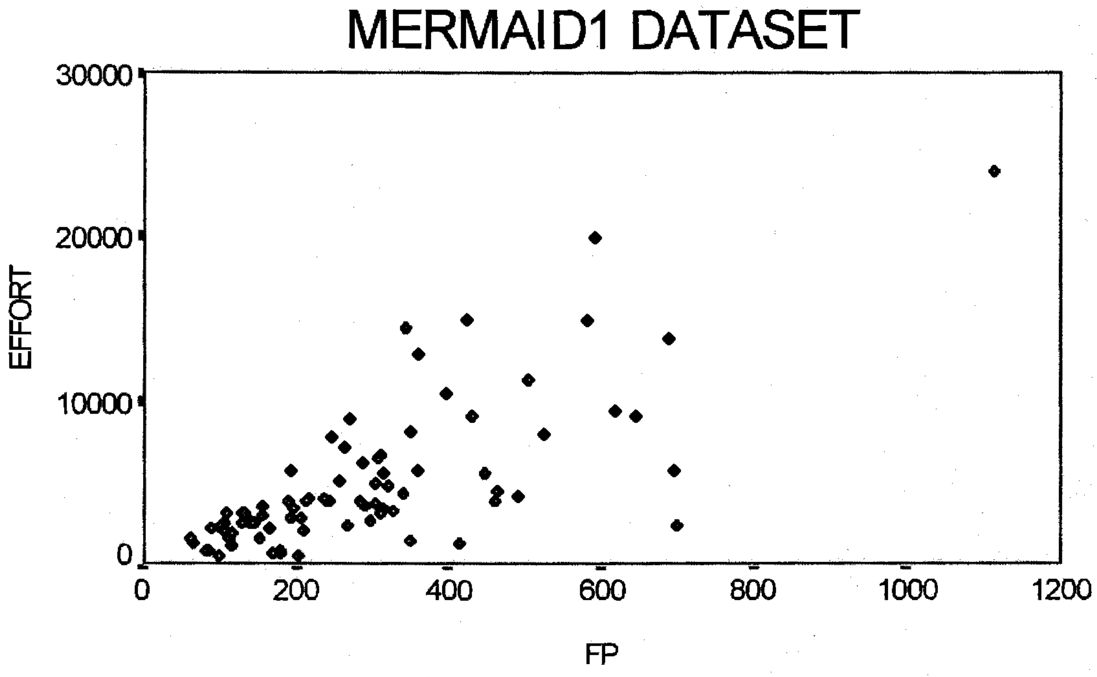

BOEHM DATASET
EFFORT
12000
10000
8000
6000
4000
2000
0
-2000
-200
0
200
400
600
800
1000
1200
LINES OF CODE

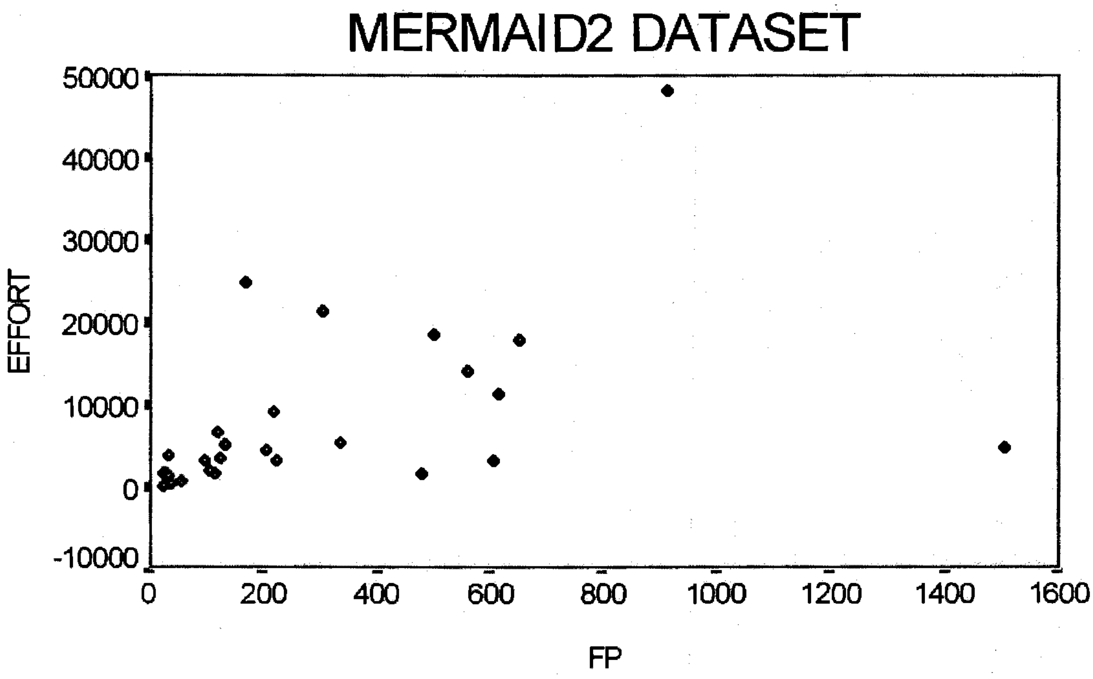

FINNISH DATASET
EFFORT
30000
20000
10000
0
-10000
0
1000
2000
FP

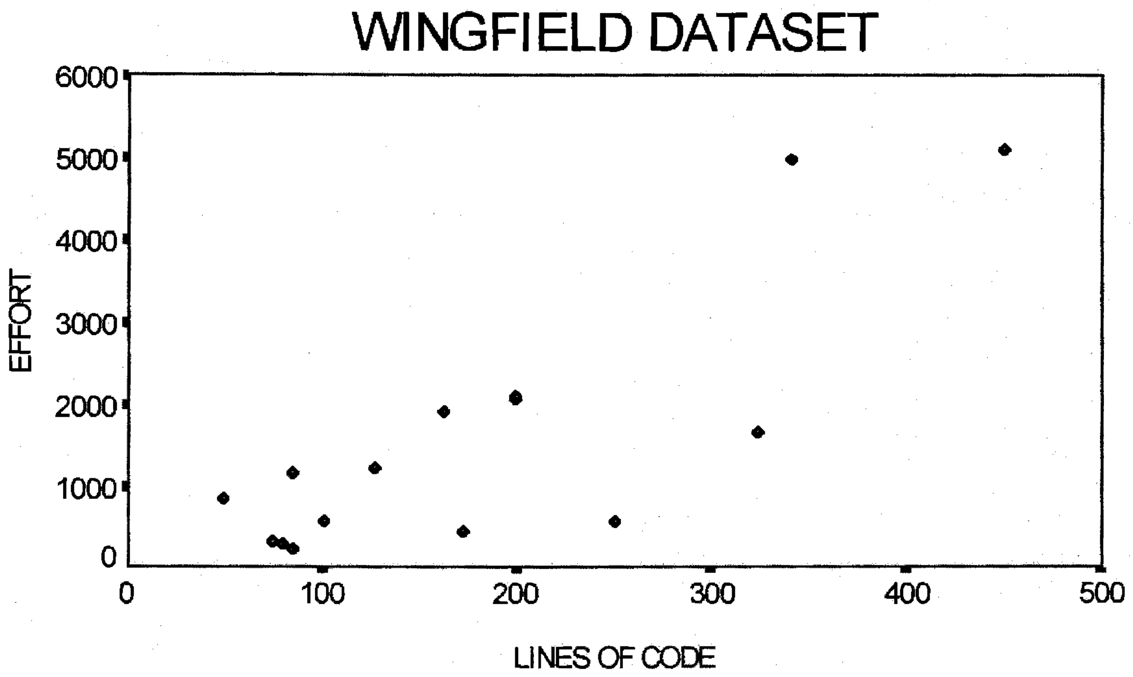
KEMERER DATASET

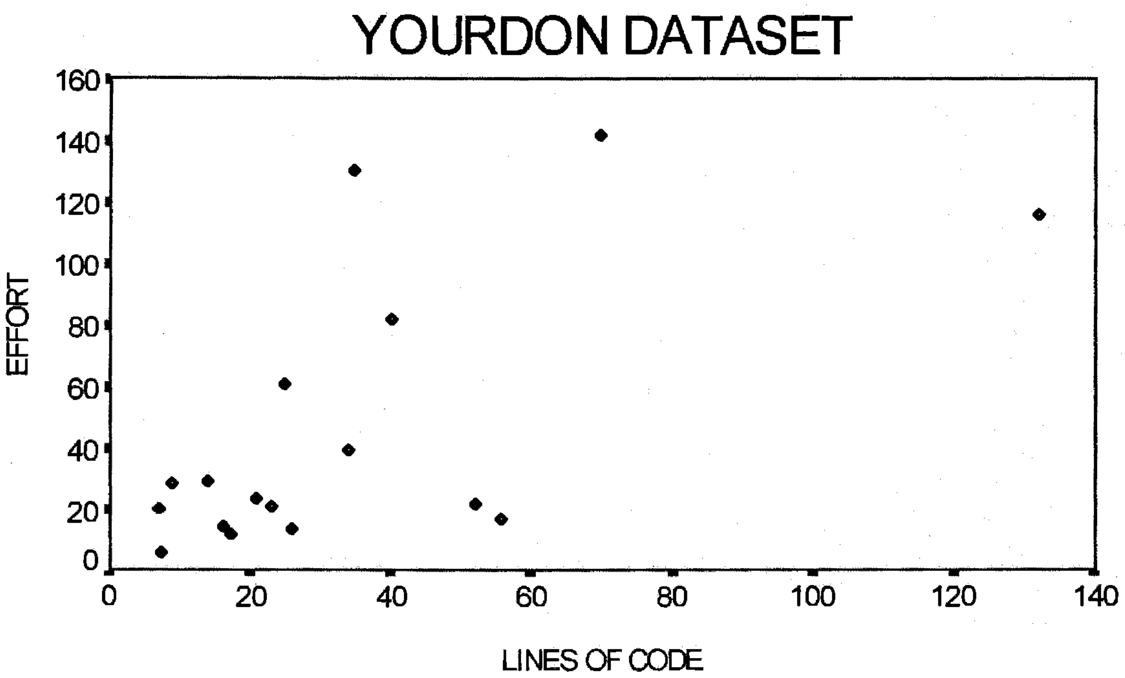



KITCHENHAM DATA
EFFORT
400
300
200
100
0
-100
0
10
20
30
40
LINES OF CODE


MERMAID1 DATASET
EFFORT
30000
20000
10000
0
0
200
400
600
800
1000
1200
FP


MERMAID2 DATASET
50000
40000
30000
20000
10000
0
-10000
EFFORT
0
200
400
600
800
1000
1200
1400
1600
FP


WINGFIELD DATASET
EFFORT
6000
5000
4000
3000
2000
1000
0
0
100
200
300
400
500
LINES OF CODE


YOURDON DATASET
EFFORT
160
140
120
100
80
60
40
20
0
0
20
40
60
80
100
120
140
LINES OF CODE

APPENDIX E

Tables are provided containing the name of the dataset and the associated variable for easy reference.

1. Linear regression

(i) Plot of residuals vs. log of the independent variable.

DATASET	INDEPENDENT VARIABLE
Albrecht-Gaffney	albfp
Belady-Lehman	BELS
Boehm (COCOMO)	adjs
Yourdon	sizey
Bailey-Basili	se
Wingfield	s1
Kitchenham-Taylor	KITS
Kemerer	kems
MERMAID-1	adfp
MERMAID-2	mer2adfp
FINN	finfp

(ii) Normal probability plots

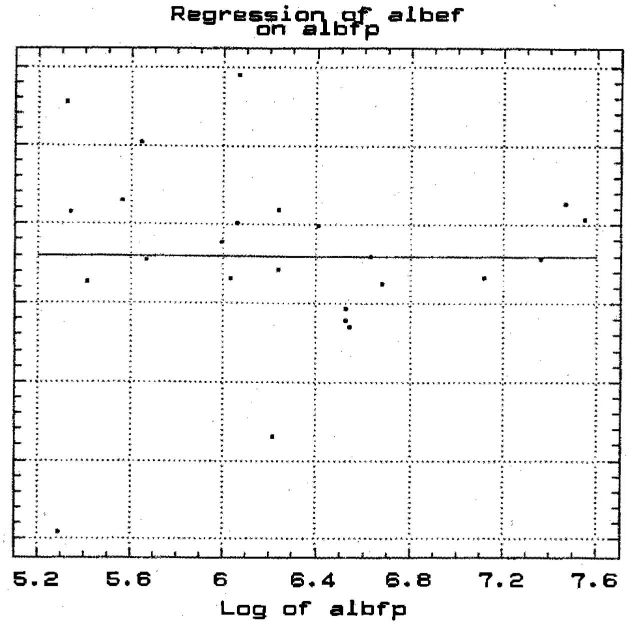
DATASET	RESIDUAL VARIABLE
Albrecht-Gaffney	resalb
Belady-Lehman	BELRES
Boehm 9COCOMO)	boeres
Yourdon	yourres
Bailey-Basili	basres
Wingfield	wingres
Kitchenham-Taylor	KITRES
Kemerer	kemres
MERMAID-1	mer1res
MERMAID-2	mer2res
FINN	finres

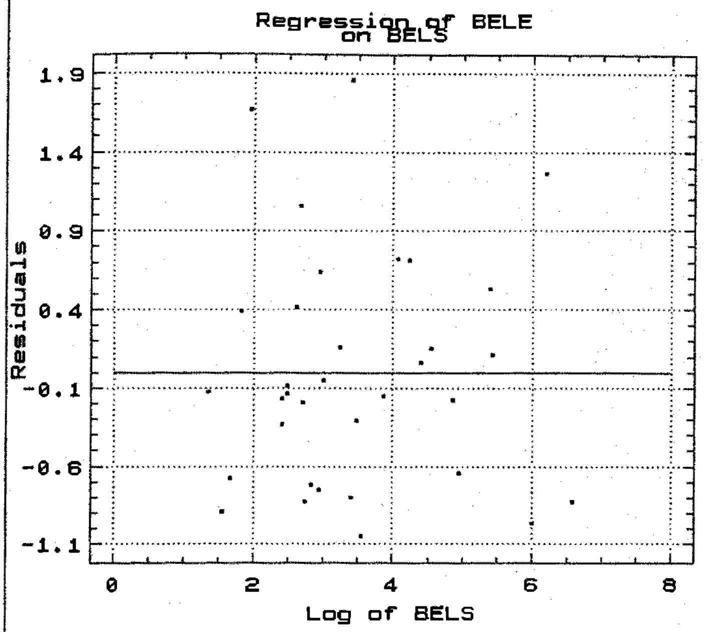
2. For the quadratic regression model

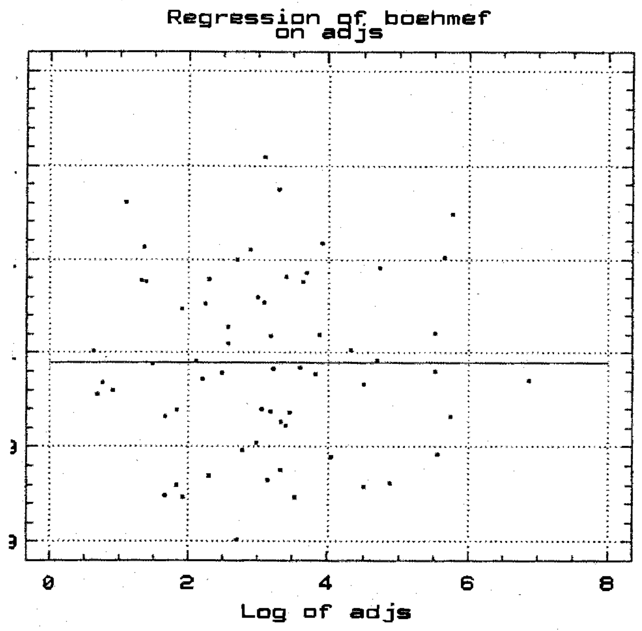
(i) Plot of residuals vs. predicted values. The variable is given on top, e.g. Residual plot for albef.

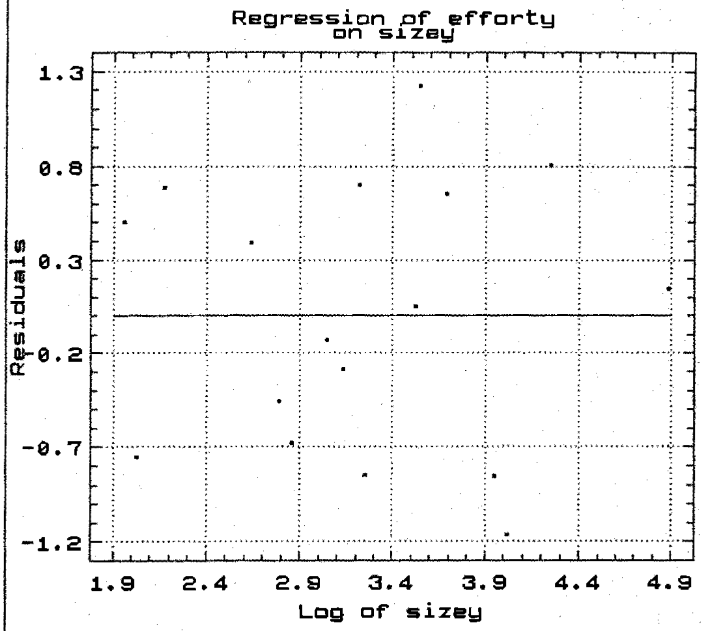
DATASET	VARIABLE
Albrecht-Gaffney	albef
Belady-Lehman	effortb
Boehm (COCOMO)	boehmef
Yourdon	efforty
Bailey-Basili	effort
Wingfield	efl
Kitchenham-Taylor	kitchef
Kemerer	kemef
MERMAID-1	meref
MERMAID-2	mer2ef
FINN	finef

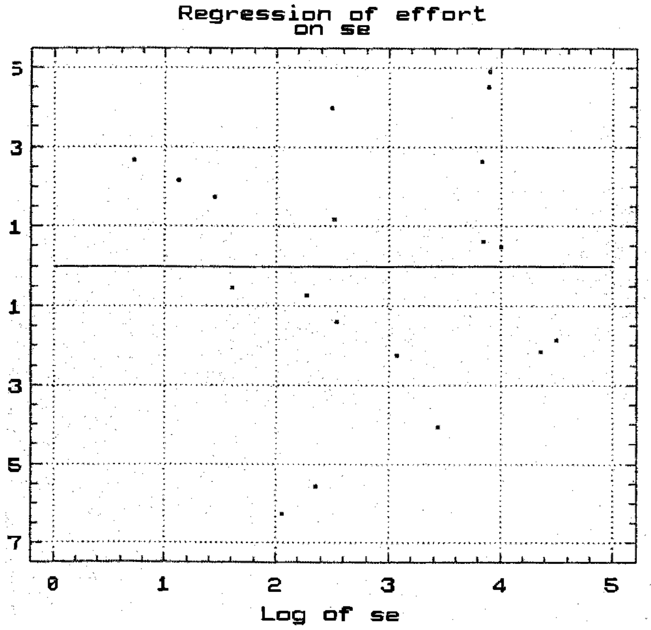
(ii) Normal probability plots

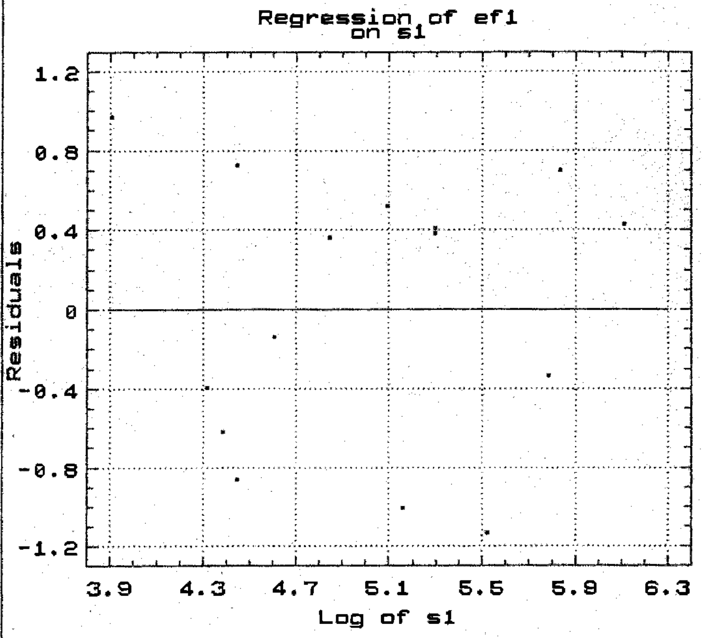

Regression of albef
on albfp
5.2
5.6
6
6.4
6.8
7.2
7.6
Log of albfp

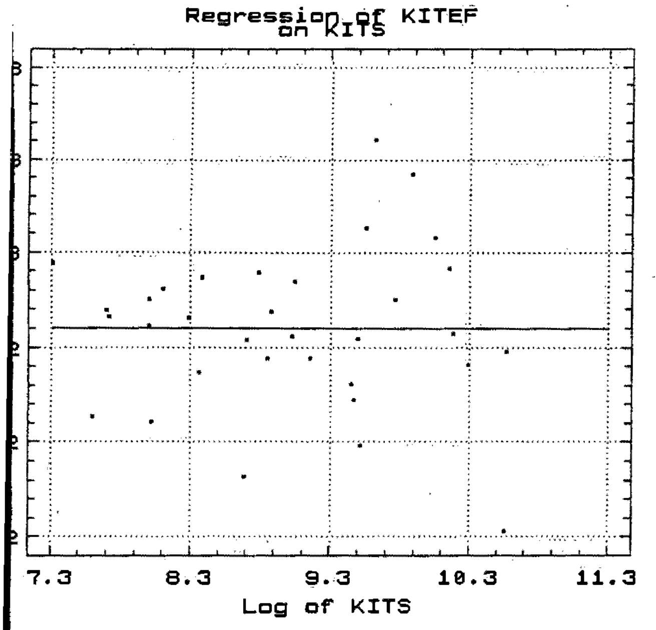

Regression of BELE
on BELS
Residuals
1.9
1.4
0.9
0.4
-0.1
-0.6
-1.1
0
2
4
6
8
Log of BELS

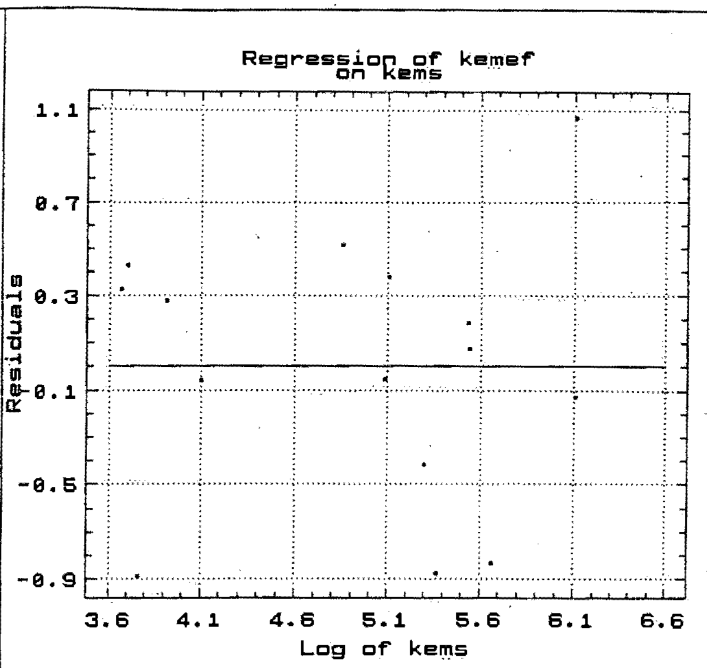

Regression of boehmef
on adjs
9
9
0
2
4
6
8
Log of adjs

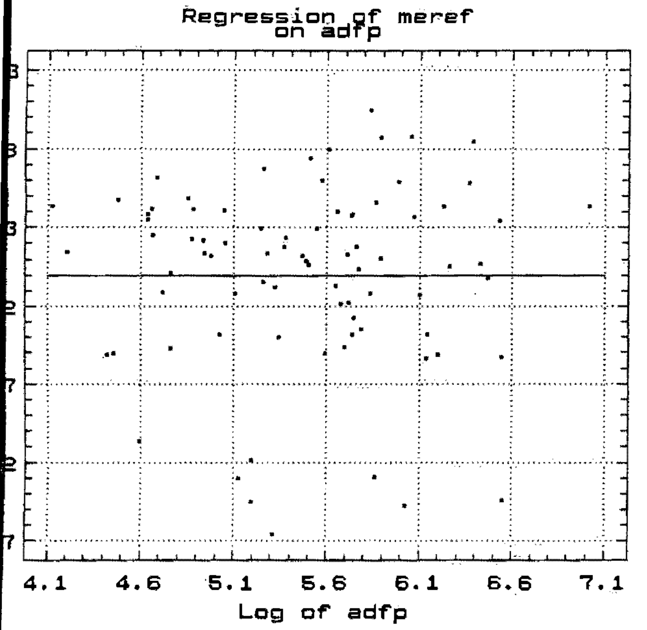

Regression of efforty
on sizey
Residuals
1.3
0.8
0.3
-0.2
-0.7
-1.2
1.9
2.4
2.9
3.4
3.9
4.4
4.9
Log of sizey

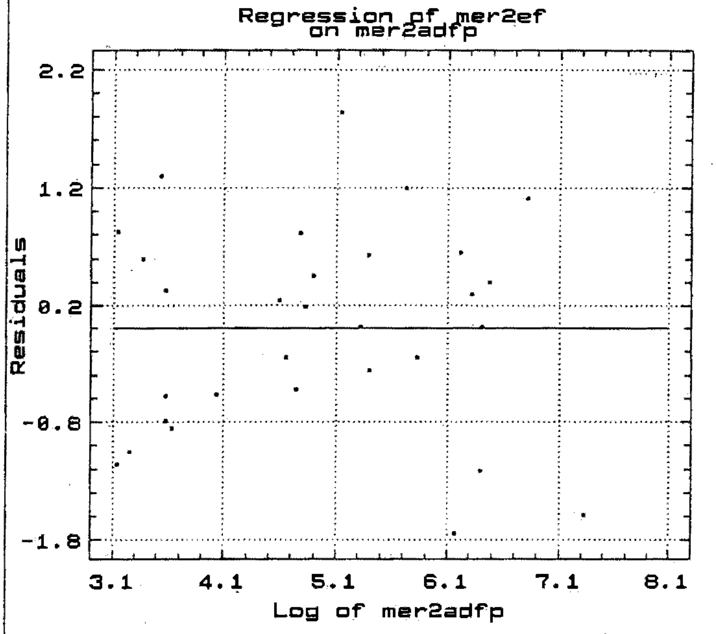

Regression of effort
on se
5
3
1
1
3
5
7
0
1
2
3
4
5
Log of se

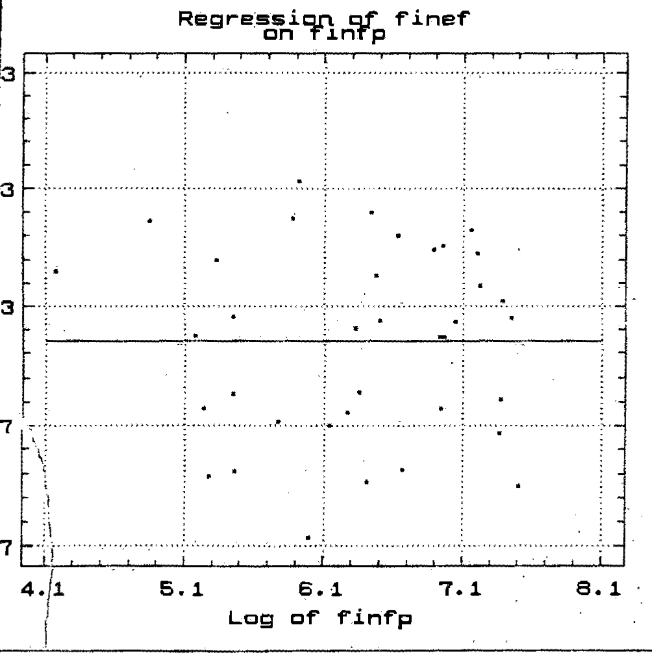

Regression of ef1
on s1
Residuals
1.2
0.8
0.4
0
-0.4
-0.8
-1.2
3.9
4.3
4.7
5.1
5.5
5.9
6.3
Log of s1

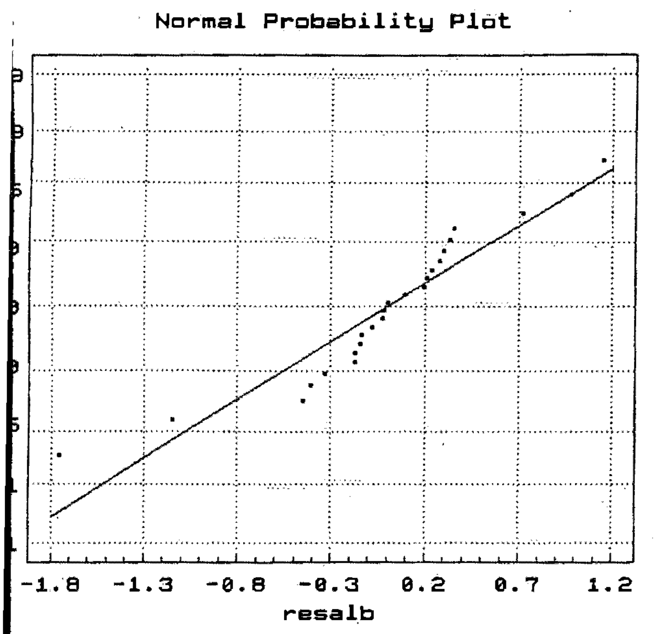

Regression of KITEF on KITS

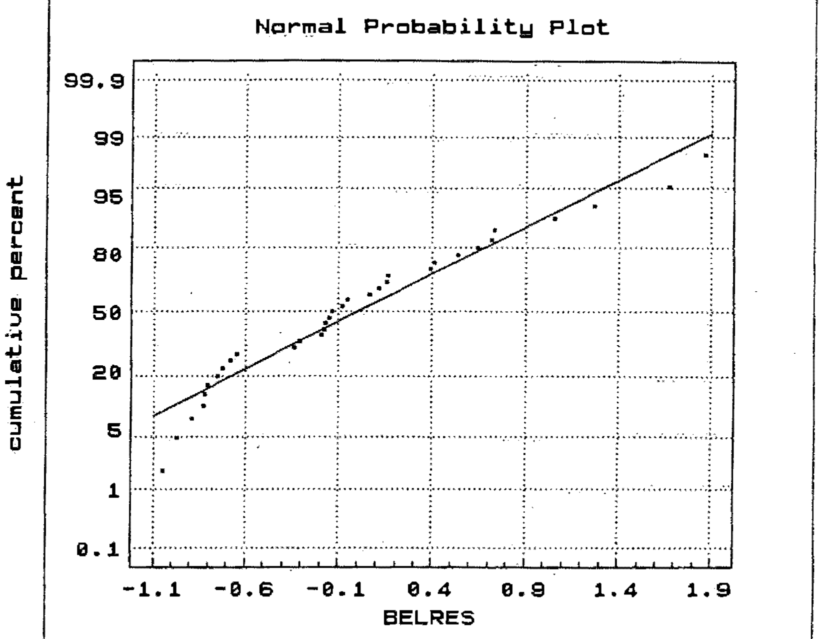

Regression of kemef on kems

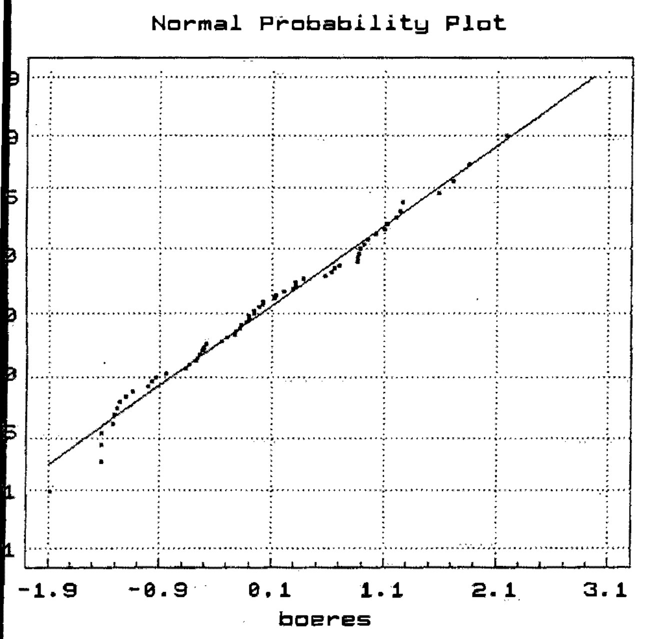

Regression of meref on adfp

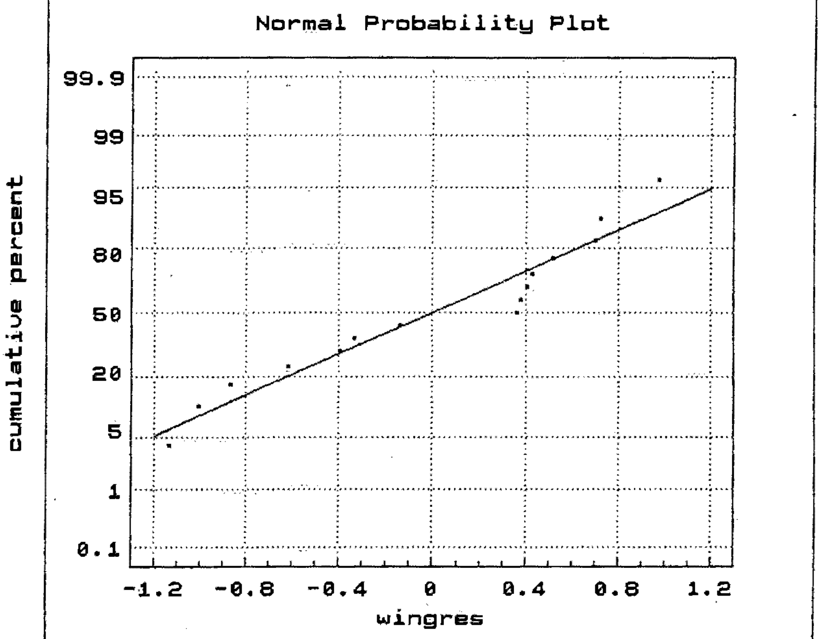

Regression of mer2ef on mer2adfp

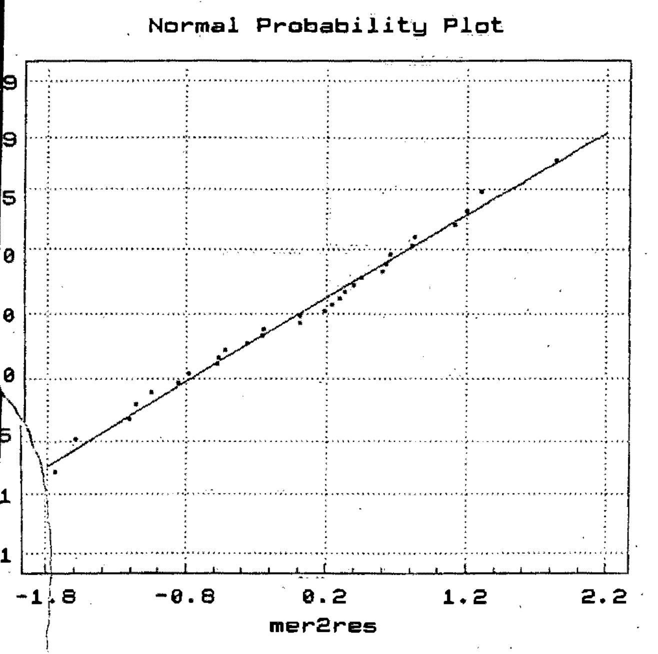

Regression of finef on finfp

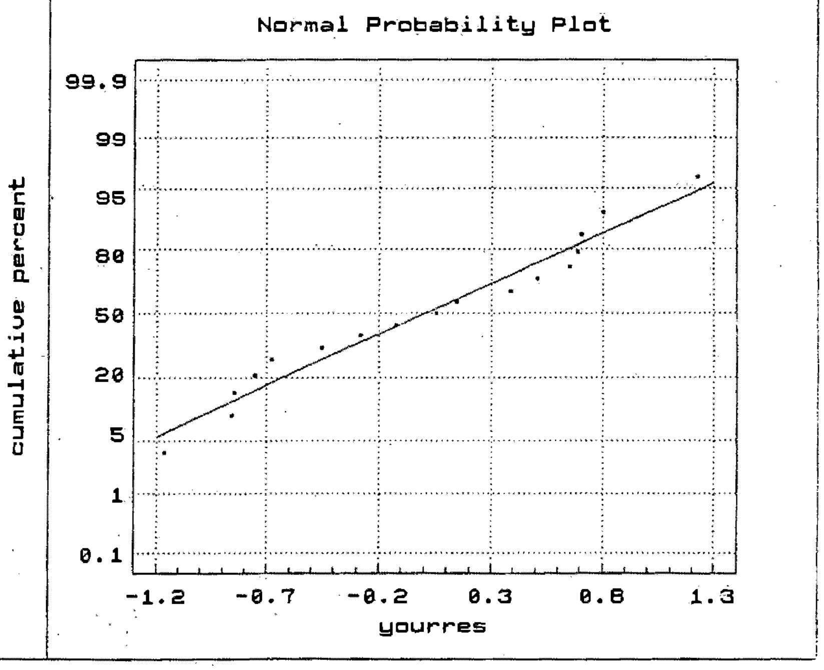

Normal Probability Plot
-1.8
-1.3
-0.8
-0.3
0.2
0.7
1.2
resalb


Normal Probability Plot
99.9
99
95
80
50
20
5
1
0.1
cumulative percent
-1.1
-0.6
-0.1
0.4
0.9
1.4
1.9
BELRES

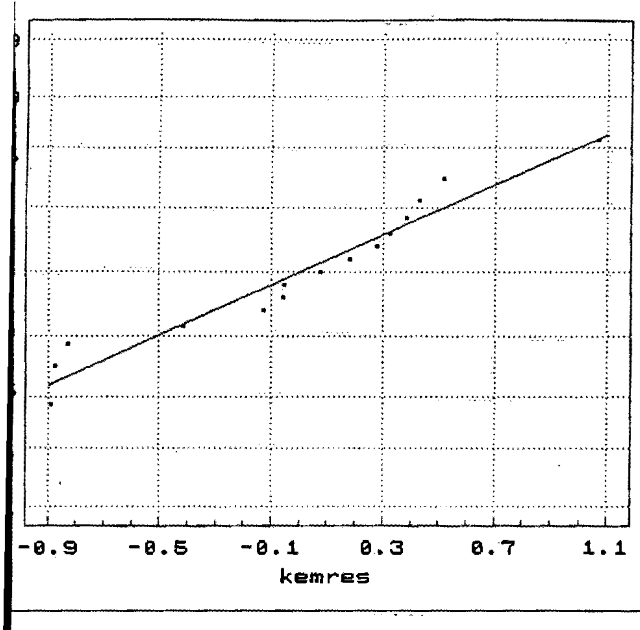

Normal Probability Plot
-1.9
-0.9
0.1
1.1
2.1
3.1
boeres


Normal Probability Plot
99.9
99
95
80
50
20
5
1
0.1
cumulative percent
-1.2
-0.8
-0.4
0
0.4
0.8
1.2
wingres

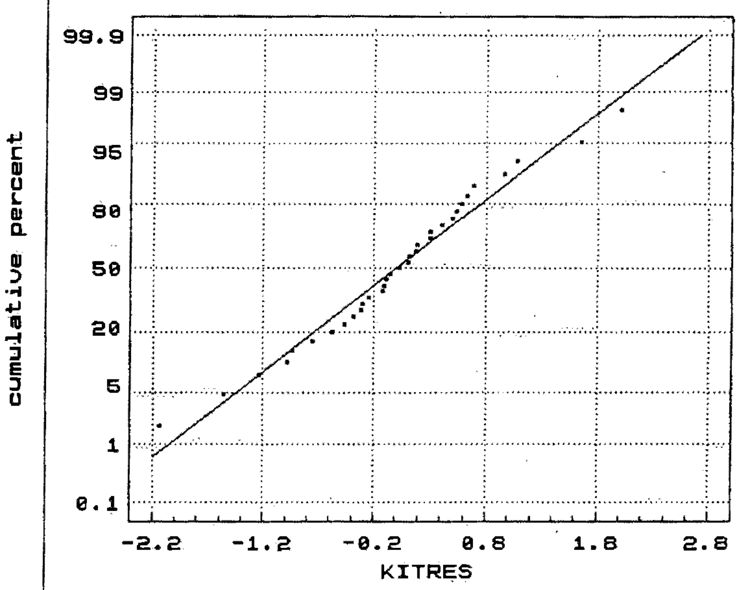

Normal Probability Plot
9
9
5
0
0
0
5
1
1
-1.8
-0.8
0.2
1.2
2.2
mer2res


Normal Probability Plot
99.9
99
95
80
50
20
5
1
0.1
cumulative percent
-1.2
-0.7
-0.2
0.3
0.8
1.3
yourres

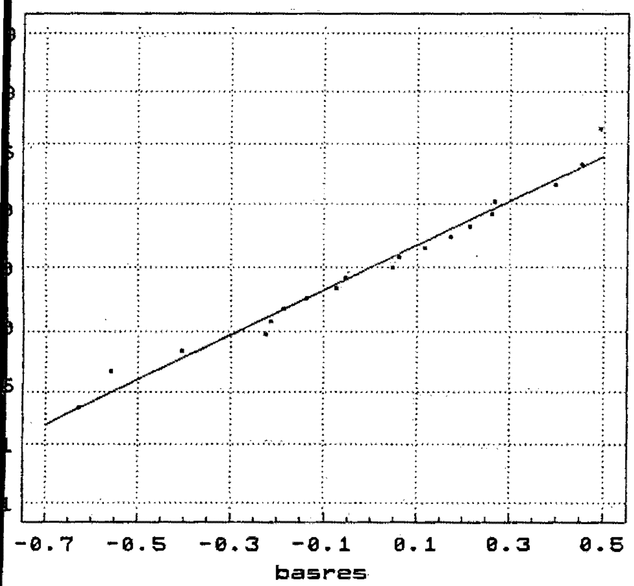
Normal Probability Plot



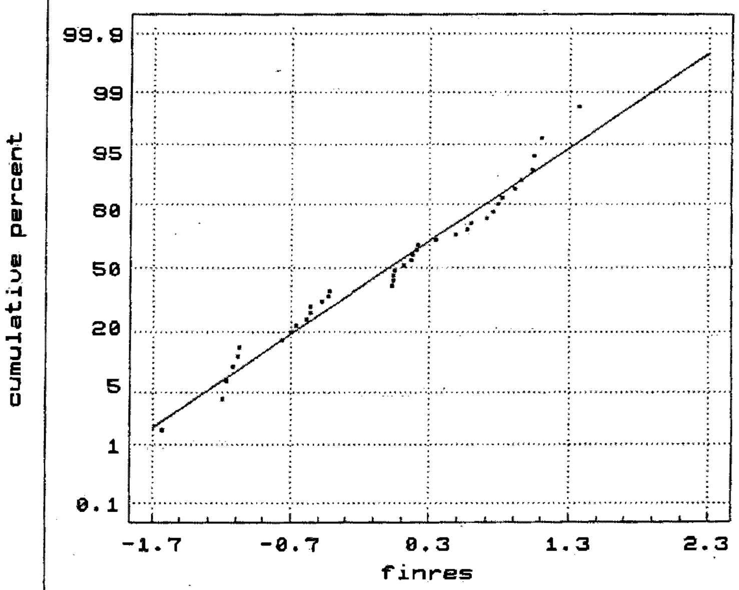
Normal Probability Plot



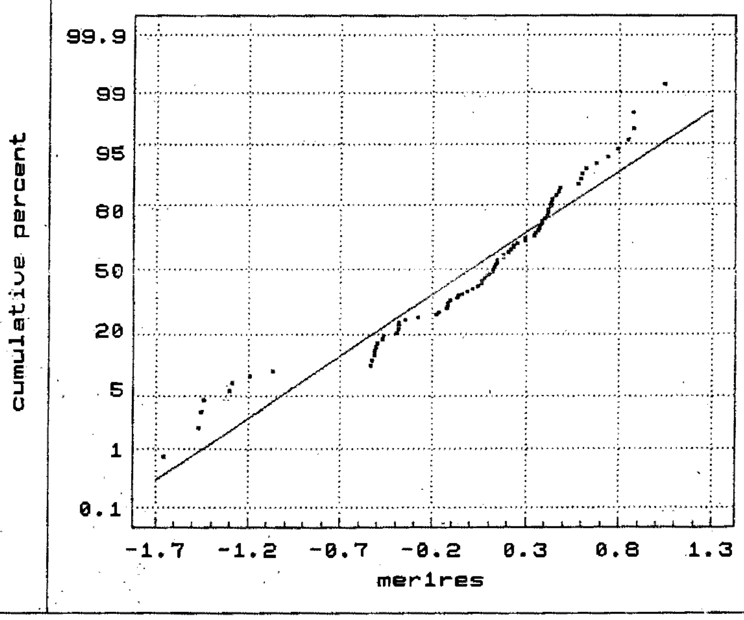
Normal Probability Plot

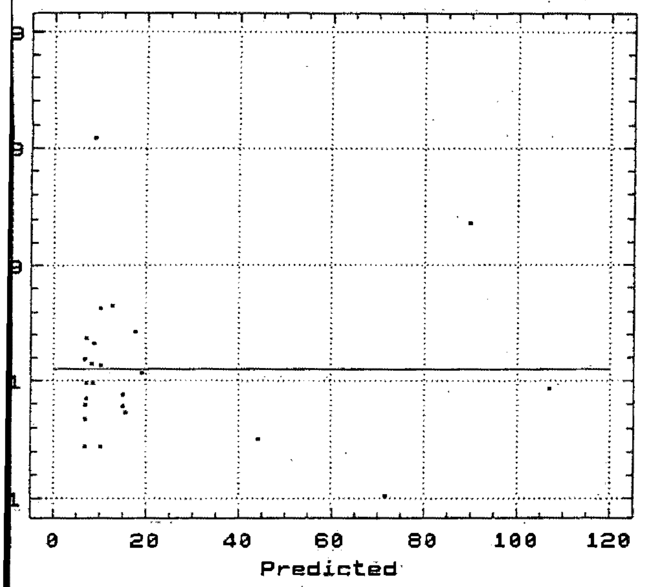


Normal Probability Plot

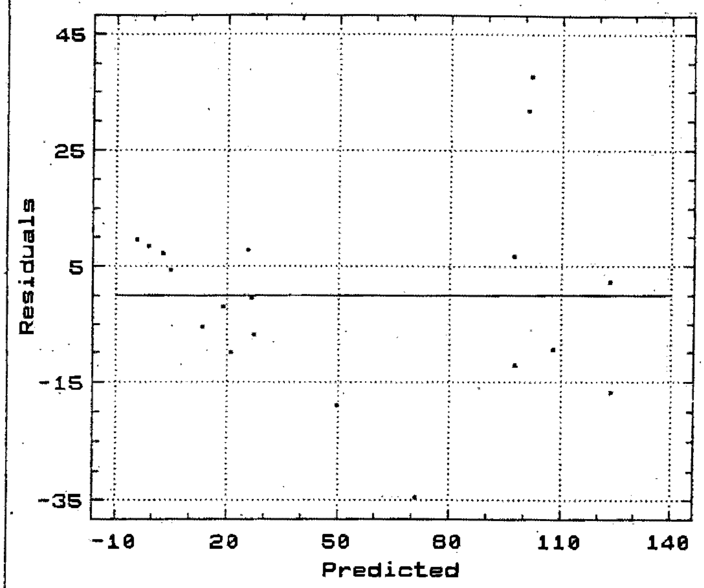


Normal Probability Plot

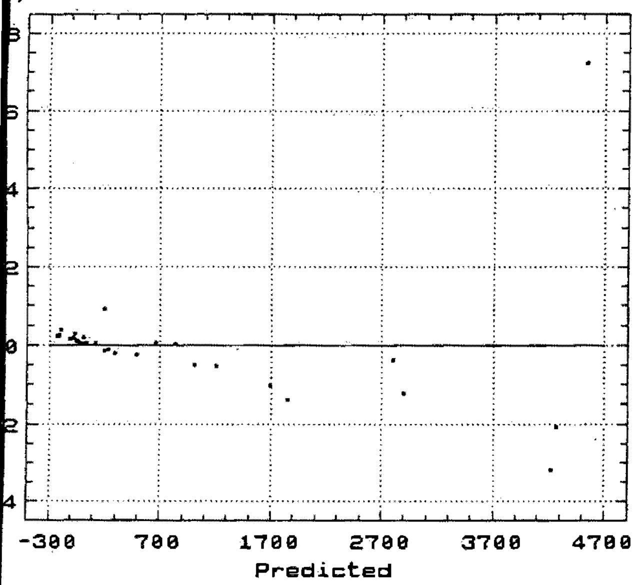




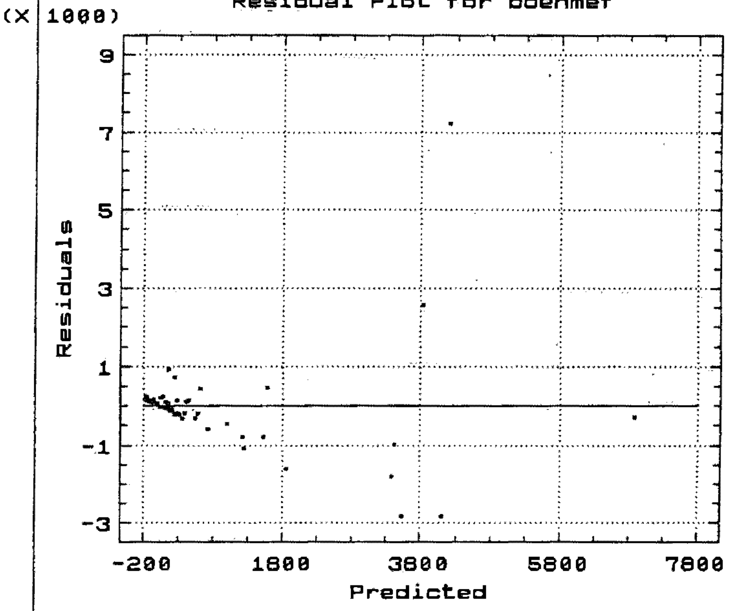
Residual Plot for albef



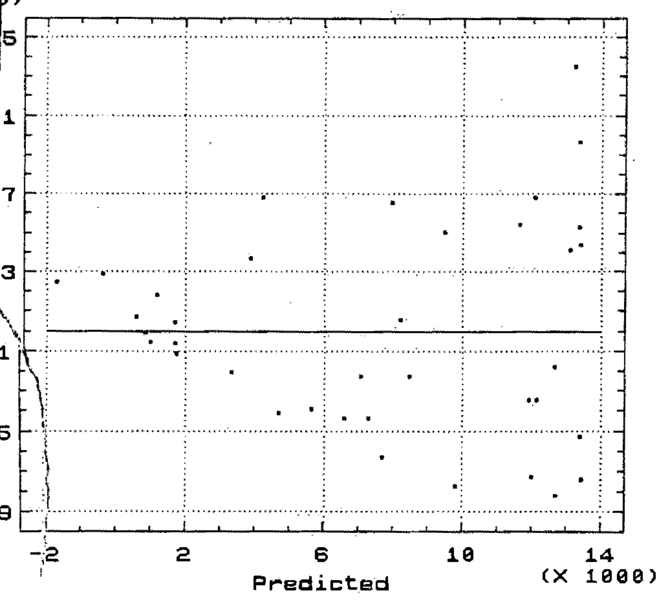
Residual Plot for effort



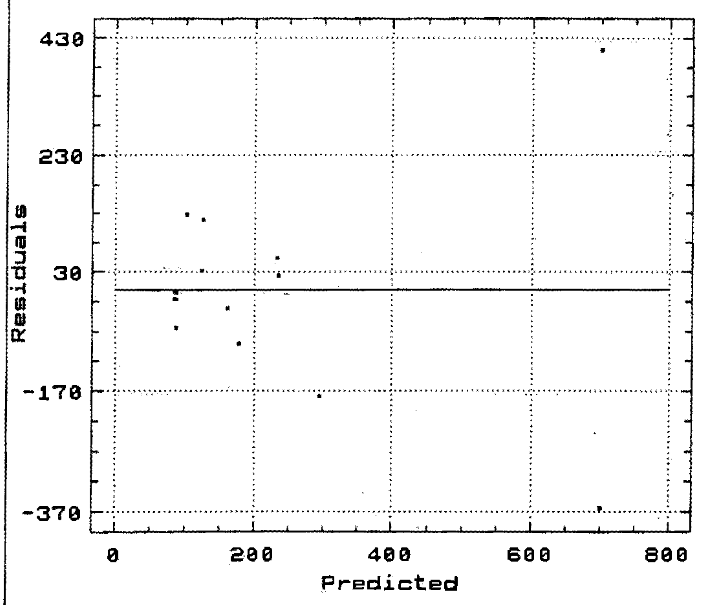
Residual Plot for effortb



Residual Plot for boehmef

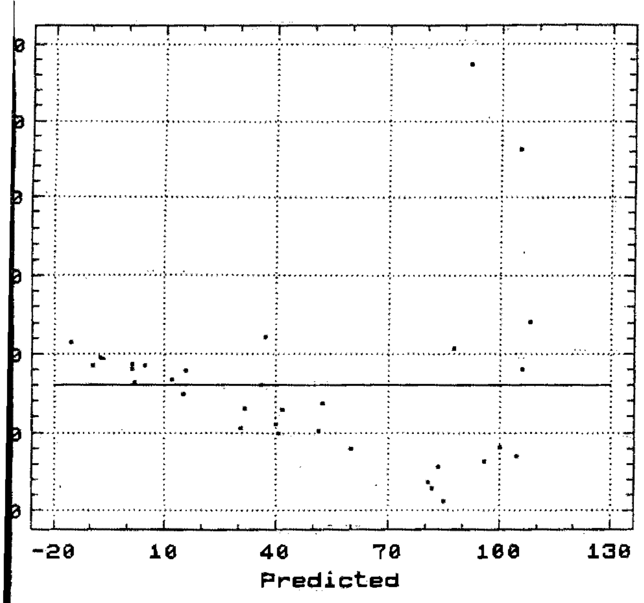


Residual Plot for finef

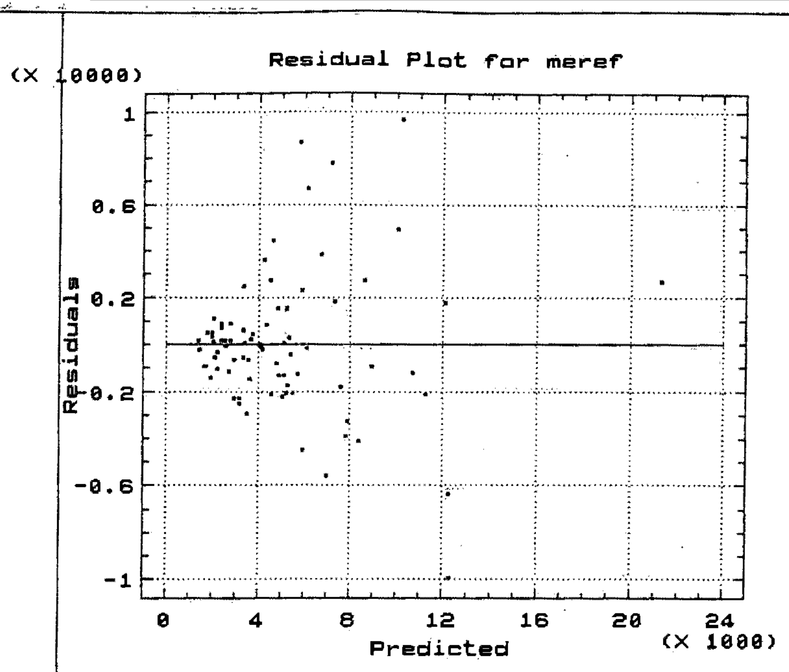


Residual Plot for kemef

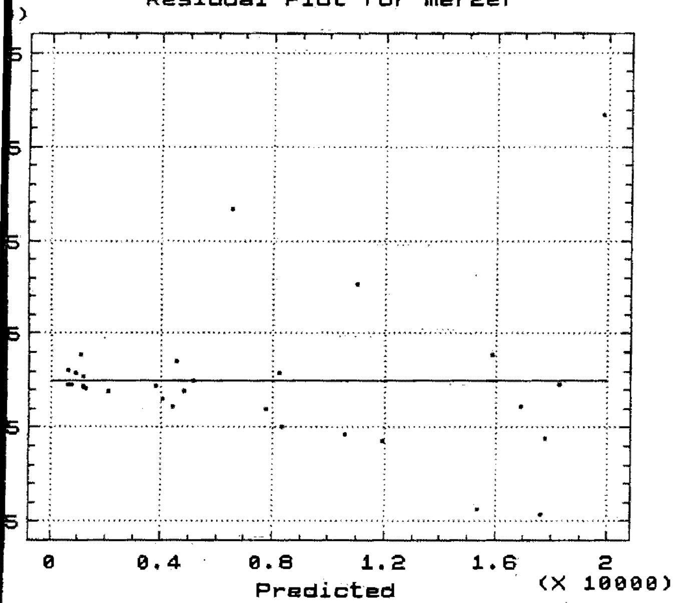
Residual Plot for kitchef



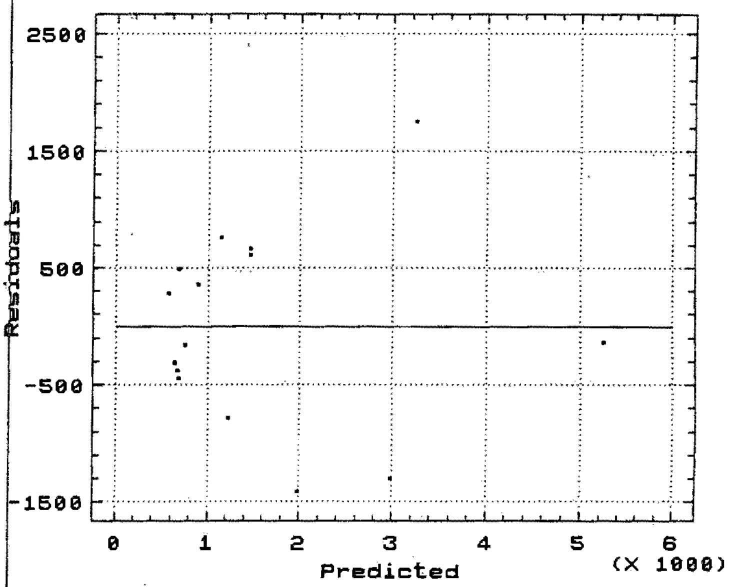
Residual Plot for meref



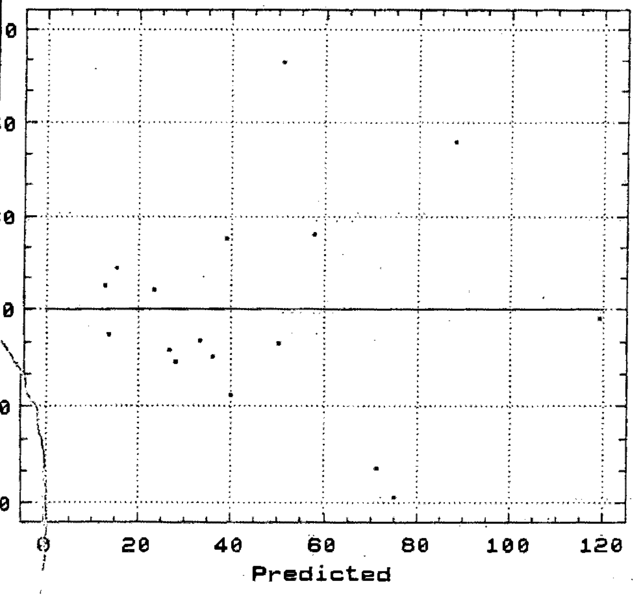
Residual Plot for mer2ef

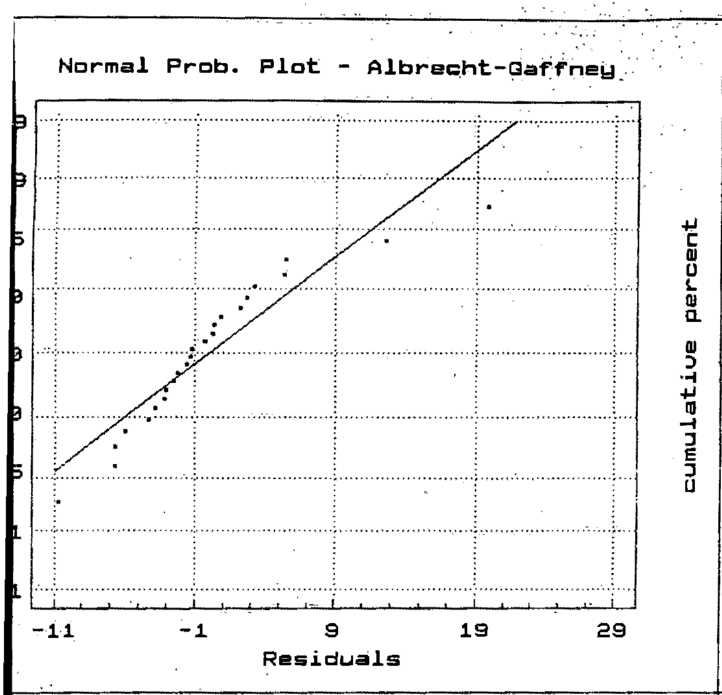


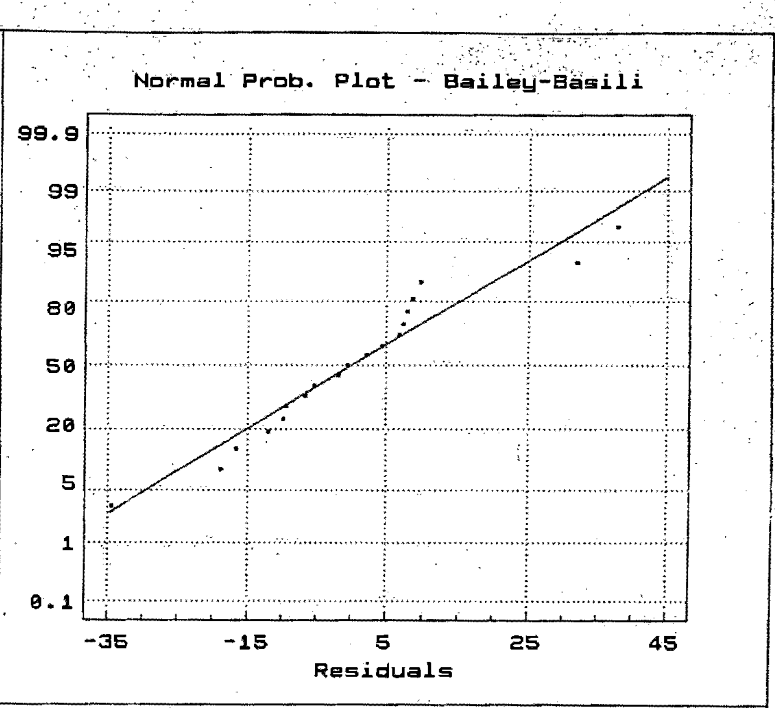
Residual Plot for ef1

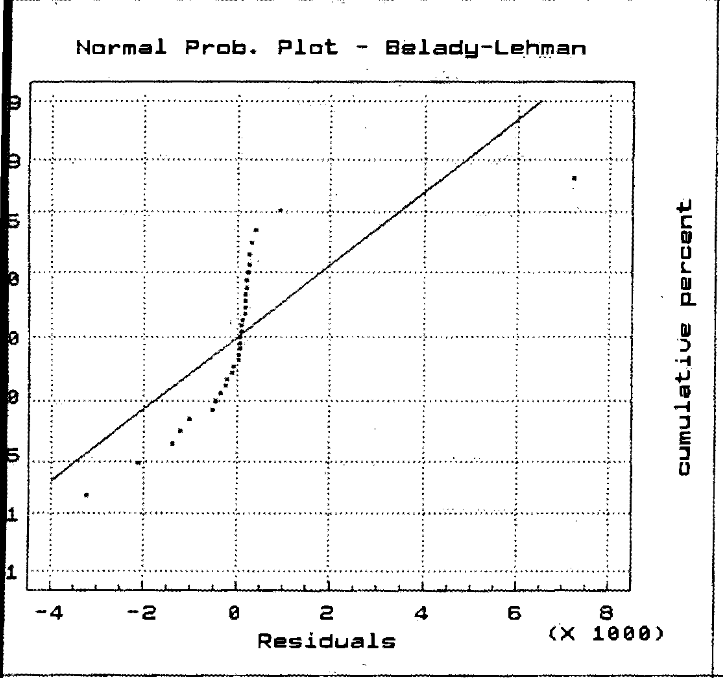


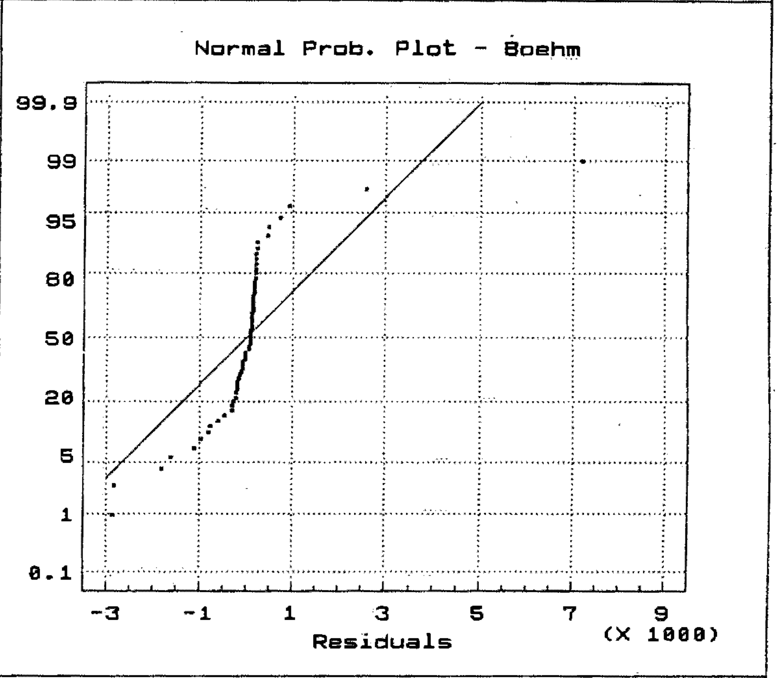
Residual Plot for efforty

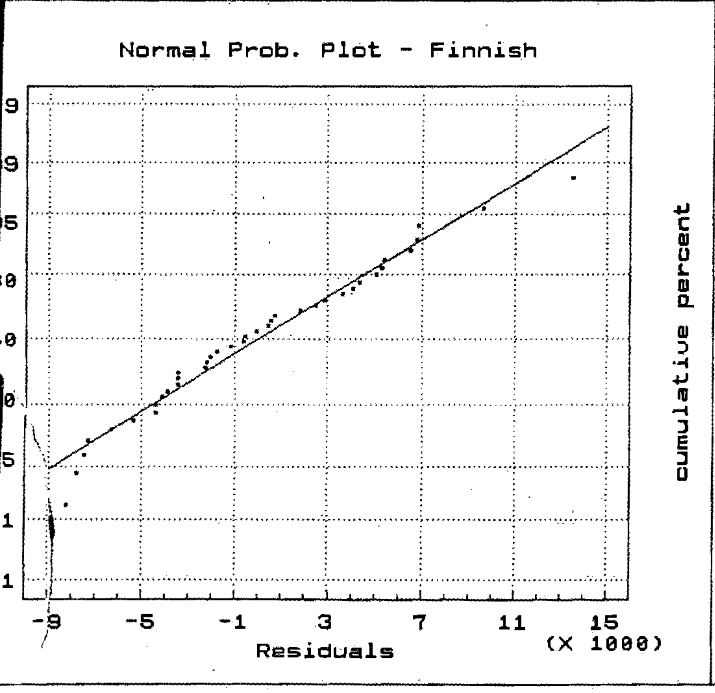


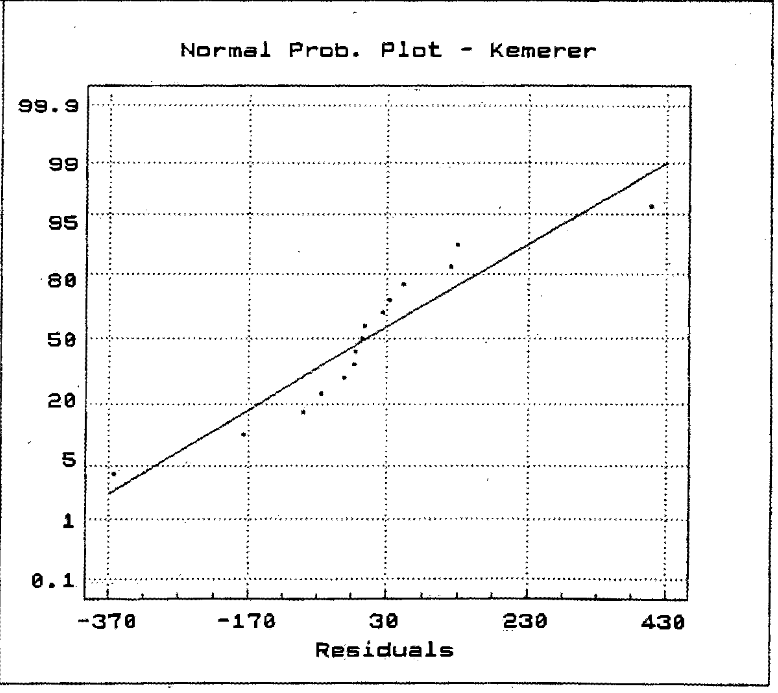

Normal Prob. Plot - Albrecht-Gaffney
-11
-1
9
19
29
Residuals

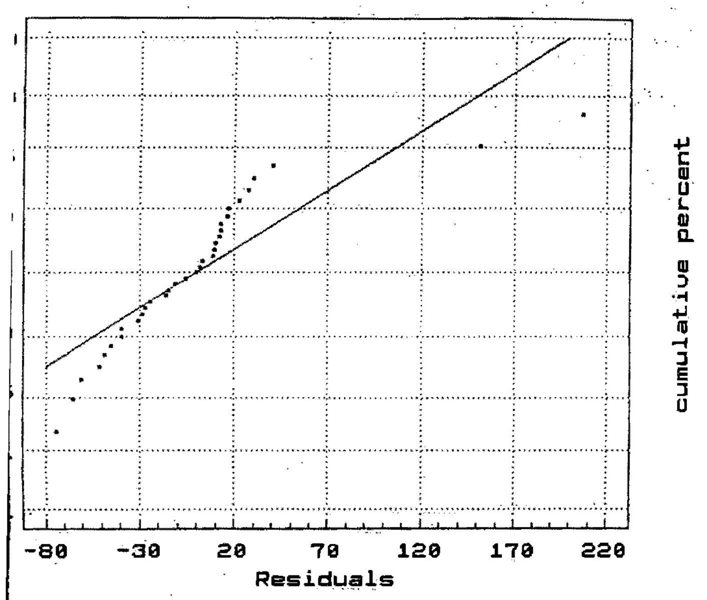

Normal Prob. Plot - Bailey-Basili
99.9
99
95
80
50
20
5
1
0.1
cumulative percent
-35
-15
5
25
45
Residuals


Normal Prob. Plot - Belady-Lehman
-4
-2
0
2
4
6
8
(X 1000)
Residuals

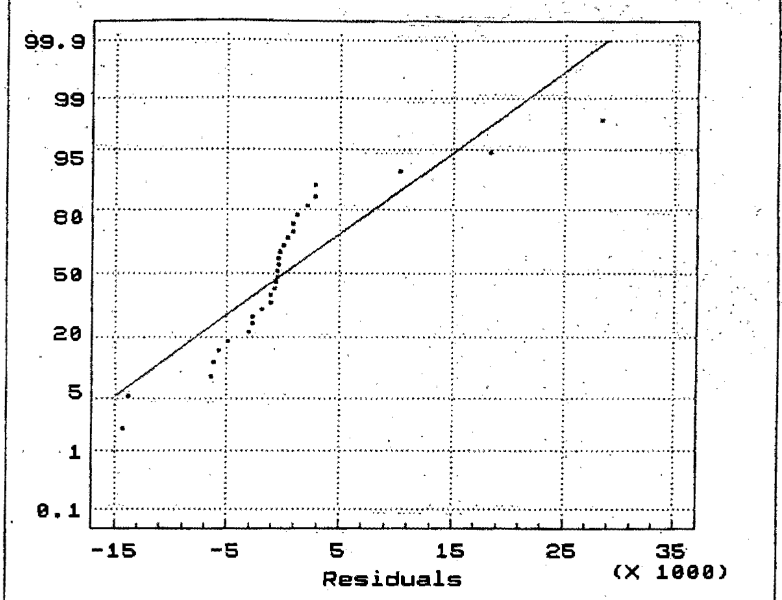

Normal Prob. Plot - Boehm
99.9
99
95
80
50
20
5
1
0.1
cumulative percent
-3
-1
1
3
5
7
9
(X 1000)
Residuals


Normal Prob. Plot - Finnish
-9
-5
-1
3
7
11
15
(X 1000)
Residuals

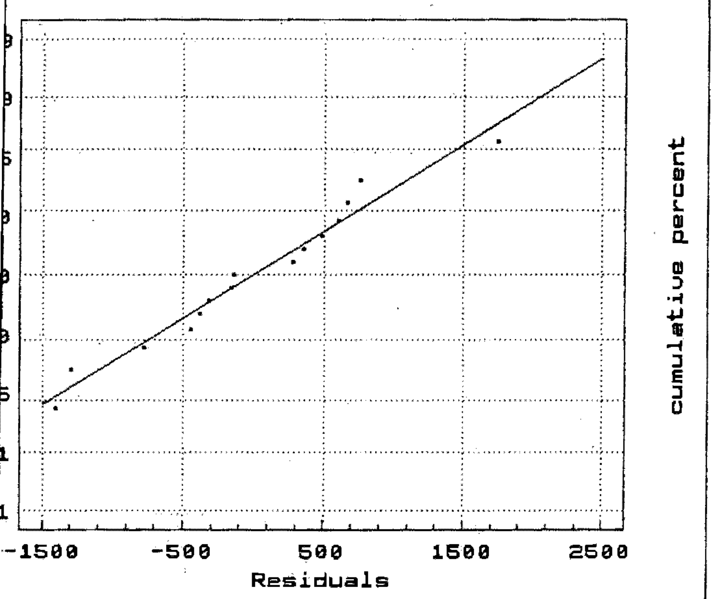

Normal Prob. Plot - Kemerer
99.9
99
95
80
50
20
5
1
0.1
cumulative percent
-370
-170
30
230
430
Residuals



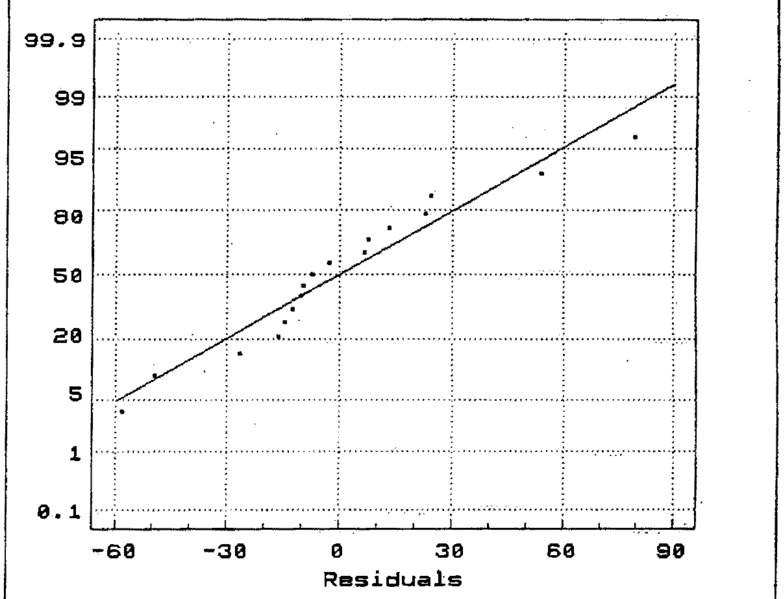
Normal Prob. Plot - Kitchenham-Taylor



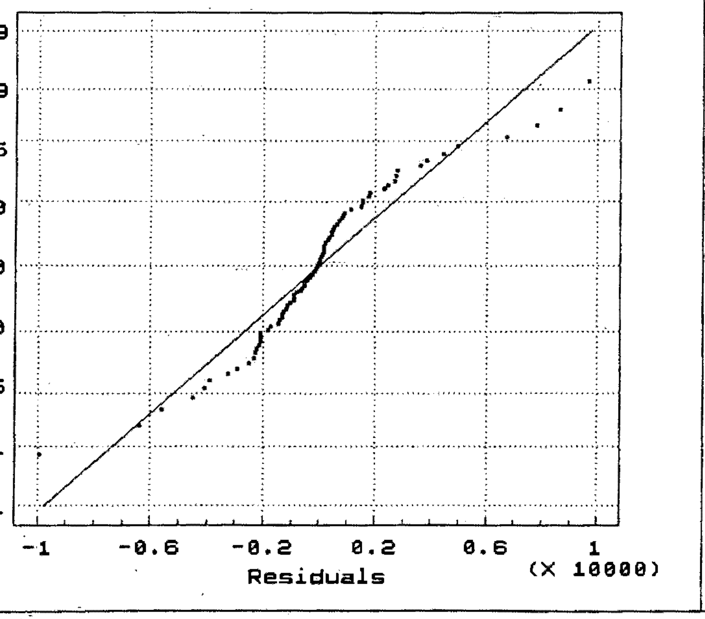
Normal Prob. Plot - MERMAID2



Normal Prob. Plot - Wingfield



Normal Prob. Plot - Yourdon



Normal Prob. Plot - MERMAID1