The Critical Success Factors of Quality Assurance and Measurement Practice in the Software Industry

Tracy Hall

Doctor of Philosophy Thesis

City University
Centre for Software Reliability

June 1998

Contents

List of tables and figures

Acknowledgements

I would like to thank Professor Norman Fenton for supervising this work. His excellent knowledge, support and enthusiasm have helped enormously.

I would also like to thank the numerous other people in the software engineering community who have discussed my work with me and who have helped me develop the ideas contained in this thesis.

Thanks also to all of the practitioners who have participated in this research.

I would also like to gratefully acknowledge the support of my employers during the completion of this thesis (Staffordshire University and the University of Westminster). I would particularly like to thank Mark Priestly, my Chair of Department at the University of Westminster for all the practical support he has provided me with, and without which completion would have been even more difficult.

Copyright Declaration

Abstract

In this thesis I analyse how QA and measurement programmes are implemented in the software industry. My thesis is that, contrary to popular belief, *how* companies implement QA and measurement is as important to the ultimate success of a programme as *what* QA and measurement tools and techniques are implemented.

I have combined input from over 300 software developers and managers in 26 different companies, with the analysis of all the relevant publicly available information on companies' experiences to produce a definitive study into implementing QA and measurement in the software industry. In this study I:

- identify those QA and measurement practices considered to be state-of-the-art;
- quantify the penetration of state-of-the-art practices;
- report on the state-of-the-practice;
- assess the effectiveness of the state-of-the-practice;
- present guidelines for improved practice.

I tested a variety of hypotheses about implementing QA and measurement programmes. My key findings about these hypotheses include:

- Some companies have a large gap between what is officially said to be implemented and what is actually implemented.

- Practitioners had a significant impact on the success of programmes. Furthermore, I confirmed my hypotheses about managers and developers having different attitudes to QA and measurement. However, I was unable to find convincing evidence to confirm that the demographics of practitioners affected their attitudes.

- Many of my hypotheses about the implementation process were confirmed. In particular I found that doing background research was related to success, but that very few companies did research. However, I was unable to find convincing evidence to confirm the importance of feedback to practitioners.

Overall my study reveals that the quality of the implementation process across the software industry is, in practice, very poor. Indeed, I show that the gap between state-of-the-art and state-of-the-practice is large. I conclude that until the software industry puts as much effort into developing strategies for implementing QA and measurement as it does into developing tools and techniques, then effective and optimised programmes will remain rare.

Chapter One

Introduction

1. The aims of the research

This research is the first comprehensive, independent attempt that I know of to review and analyse current QA and measurement practice in the UK software industry. My objective is to use the data I collect during this review to identify the critical success factors for QA and measurement programme implementation in software development environments. To this end I have used the data collected during this study to design a summary of recommendations which guides software companies on the effective implementation of QA and measurement. I aim to show that by controlling critical success factors during the implementation process, programmes can be managed to optimise quality.

1.1 How and what to implement

I believe that programmes must be analysed both in terms of *what* QA and measurement practices should be implemented, and *how* they should be implemented. The rigor and validity of a programme's overall implementation and management framework is as important as the rigor and validity of the individual practices within a programme. Most other studies focus only on *what* to implement, and although the validity of individual practices is important, it is irrelevant if a new practice is within such a poor QA and measurement framework that practices are not used. I focus on *how* software companies can successfully implement QA and measurement programmes.

1.2 Managing the implementation process

I primarily concern myself with the organisational and management aspects of implementing and sustaining effective QA and measurement programmes in commercial software development environments. I believe that whether individual quality practices are technical (for example, introducing CASE) or organisational (for example, applying the Capability Maturity Model [Humphrey 1989]), success is dependent on managing the implementation process effectively.

1.3 Studying industry practice

I have conducted an extensive study of the software industry to identify and compare current *best* QA and measurement programme practice and current *normal* practice. In doing so I get behind the QA and measurement 'hype', to find out what is actually *viable*, *effective* and *practical* at grassroots in ordinary software companies. I confirm a significant gap between the state-of-the-art and the state-of-practice. [Card 1988] first speculated on such a gap in 1988.

1.4 Analysing the literature

Despite software QA and measurement appearing to be discrete topics, when I analysed the literature the majority of the issues surrounding *the implementation* of QA and measurement are similar. Although some of the implementation factors are reported as QA or measurement specific, most are equally relevant to the successful implementation of either QA or measurement. The only difference is the actual tools, techniques and methods being implemented. I have taken the view that measurement is simply one of many QA tools that can be used to help improve quality.

There has been a great deal of work done in developing individual QA and measurement practices and methods (for example [Kitchenham 1984]), and to a lessor extent, in measuring the industrial penetration of those practices [Drummond-Tylor 1989]. However, there is very little work reported which describes how to successfully introduce and manage QA and measurement programmes.

1.5 Reviewing other studies

Although a few studies with similar aims to mine have already taken place, they have suffered from the following limitations:

1. Emphasis on *what* to implement rather than *how* to implement.
2. The restriction of data to the experiences of single companies.
3. Data that is biased.
4. The limited availability of data.
5. Restricted accessibility to data.

I discuss each of these limitations in more detail below:

- **Emphasis on *what* to implement rather than *how* to implement**

  Although my study has similar aims to existing work, other studies have focused exclusively on QA and measurement tools and methods. A good example of such work is the development of the Capability Maturity Model (CMM) [Humphrey 1989] and the People Maturity Model [Curtis *et al* 1995] by the SEI. My research is distinct, as I focus primarily on:

  - the initial implementation and sustenance of QA and measurement;
  - the organisational and management aspects of QA and measurement;
  - the contribution and impact of practitioners† to the success of QA and measurement.

† I use the term 'practitioner' when no distinction has been made between managers or developers, nor between technical domains

- **Data that is restricted to single company experiences**

  Most earlier studies have only documented the QA and measurement practices of single companies. This means that the data reported has minimal value to the industry as a whole. An exception to this is [Jeffery & Berry 1993], which reports on four companies. I will discuss this and other studies in more detail in Chapters Three and Four.

- **Data is biased**

  Most published data is exclusively positive. Most reports are written by company employees who portray their company's programmes as both successful and straightforward to implement (despite reports that up to 80% of programmes fail [Kearney 1992]). However, some reports are more balanced and report failures as well as successes (for example, [Jeffery & Berry 1993, Lindstrom 1993]).

- **Limited availability of data**

  Objective public domain data that tells us how companies are implementing QA and measurement is in renowned short supply in software engineering [Hetzel 1995, Fenton *et al* 1994]. Consequently, many other reports suffer from a restriction on what data they can use and publish. There may be a variety of reasons for such non-publication, including that:

  - the introduction was not a success and companies do not want to admit it publicly;
  - the introduction was a success and companies do not want to lose the competitive edge that this has given;
  - they do not want to publish any data that will reveal the real quality of their products (Motorola is an exception to this [Weller 1993]).

- **Restricted accessibility to data**

  Some studies have been conducted which claim to have collected data representative of the whole industry. Unfortunately, to my knowledge, all of these have been done by consultancy companies. This means that their data is rarely in the public domain. It also means that the cost of gaining access to any reports from these sources is usually prohibitive. Moreover, the scientific basis of this type of research is not always transparent.

1.6 The uniqueness of this work

My study is unique as I report on the QA and measurement data that I have collected from twenty-six different companies. Furthermore, the data that I have collected is both quantitative and qualitative. The large quantity of high quality data that I have collected makes the results and conclusions of my study significant, and, therefore, of great value to the software community.

During the study I used two methods of data collection:

- **Academic Research**
  I analysed all the available published company QA and measurement case studies. I also consulted many QA and measurement promoters and commentators. My aim was to identify the current QA and measurement state-of-the-art.

- **Surveying Techniques**
  I observed QA and measurement in twenty-six companies from a cross-section of the UK software industry. I used practitioners themselves as my primary data source and in doing so I received input from over three hundred software practitioners. A wide range of practitioners were involved in the study, from senior corporate managers to graduate programmers. My aim was to identify both the *officially claimed* and *actual* QA and measurement state-of-practice within typical software companies.

Using these two data collection methods I was able to identify a significant gap between the state-of-art and the state-of-practice. I also discovered a gap between what companies said they were doing in terms of QA and measurement and what they were actually doing. Furthermore, I found that the validity and viability of some aspects of the QA and measurement state-of-the-art did not always stand scrutiny.

2. The context of the research

2.1 The software industry

Mature and well established industries, such as engineering, have over time developed integrated, effective approaches to QA and measurement. In such industries standard QA and measurement practices are broadly accepted by practitioners, and consequently are implemented on an industry-wide basis. The software industry is still relatively immature, and as such its QA and measurement practices tend to be immature and *ad hoc*. As yet there is no industry-wide consensus on how QA and measurement should be tackled, and although some companies have implemented sophisticated and effective programmes, most have not. Indeed, overall, software companies have been slow to implement QA and measurement. DeMarco comments very appropriately on the current situation regarding measurement:

> *"Real measurement is hard work and it might point up some scary results, so many organisations simply don't do it. Their philosophy is hear no evil, see no evil, speak no evil"* [DeMarco 1995]

2.2 Software quality

Much has been written about the poor quality of software still produced by the industry and quality continues to be a major problem [Flowers 1996]. Furthermore, it is a problem widely acknowledged to be a management rather than technical problem [Flowers 1996, Iizuka 1995, Jones 1994].

I believe that for a major shift in software quality to take place, two industry-wide pre-requisites are needed:

1. **Consensus:** There needs to exist a broad consensus on what constitutes QA and measurement best practice.

2. **Practice:** There needs to be a basic level of QA and measurement actually practised.

Both of these pre-requisites are now being addressed by various international bodies. Indeed, the development of models such as CMM [Humphrey 1989], SPICE [Konrad *et al* 1995] and TickIT [Dti 1992], has led to an increased consensus on QA and measurement best practice. Similarly, industry penetration of basic QA and measurement also seems to be increasing. However, despite this general improvement I found evidence to suggest that there was greater consensus about software measurement best practice than QA best practice. There was more ambiguity, confusion and conflict amongst practitioners as to what constituted QA best practice.

It is only by identifying a *demonstrably* effective base set of QA and measurement practices, that consensus and good practice can grow. In this research I aim to identify the QA and measurement practices that are most likely to be successful to the greatest number of software development companies and so contribute to a consensus of best QA and measurement practice.

3. The nature of the research

There is very little evidence of practitioner views being sought or incorporated into either industry-wide or company-specific quality initiatives. Yet it is my view that the attitudes of individual practitioners directly influence software quality. The attitudes that individual practitioners have to software quality, and to the use of specific quality mechanisms, will dictate that practitioner's behaviour towards quality. This behaviour will, in turn, have a direct impact on the quality of the software he or she produces. The impact of attitude on behaviour is more readily accepted in social science disciplines than in software engineering [Oppenheim 1992]. The impact of practitioner attitude is frequently overlooked during the development of new quality initiatives. This probably contributes significantly to the failure of many quality initiatives. I assert that understanding the attitudes of practitioners to quality is a precursor to implementing successful quality initiatives.

This research is unique as I collect data measuring *practitioner* experiences and perceptions of QA and measurement. I concern myself with what practitioners say about the implementation of QA and measurement. Although there is anecdotal evidence about practitioners' views and experiences of QA and measurement, in this research I formally collect this 'anecdotal' data.

4. Research objectives and hypotheses

4.1 Objectives

My aim with this research was to identify the critical success factors of implementing QA and measurement in the UK software industry. In particular I aimed to:

- identify those QA and measurement practices considered to be state-of-the-art
- quantify the penetration of state-of-the-art practices
- report on the state-of-the-practice
- assess the effectiveness of the state-of-the-practice
- present guidelines for improved practice

4.2 Thesis and hypotheses

4.2.1 Thesis

My thesis is that QA and measurement can be implemented in such a way to make it possible to manage and control for a successful programme.

4.2.2 Hypotheses

I also had various low level hypotheses that I tested during the research. I looked particularly for evidence to support the following.

4.2.2.1 Hypotheses about the company

i QA and measurement are most successful when a quality culture exists within the company.

ii QA and measurement is more likely to be successful where morale in a company is already quite high and practitioners generally feel positive about the company.

iii It is common for companies not to practice what they claim to in terms of QA and measurement.

iv Companies which have highly structured and disciplined existing working practices are more likely to have effective QA and measurement.

v QA and measurement that is implemented ineffectively will have widespread counterproductive repercussions throughout the company.

Summary of Findings

Chapters Seven and Eight reveals the results of my research relative to these hypotheses. Notably, I found particularly strong evidence for hypothesis iii. Conversely, I could find no evidence to support hypothesis ii; in fact some companies with low company morale had effective programmes while other companies in which I measured high morale had an ineffective programme. Some hypotheses proved difficult to provide much evidence for or against. For example, hypothesis v was too general to collect enough data on which to reject the null hypothesis.

4.2.2.2 Hypotheses about the practitioners

i Practitioners' attitudes affect the success of QA and measurement.
ii Managers and developers have different views on which QA and measurement practices are worthwhile.
iii Managers are enthusiastic about QA and measurement.
iv Developers resist the implementation of QA and measurement.
v QA and measurement does not allow practitioners to be creative.
vi Younger practitioners who are more highly qualified will be keener on QA and measurement.
vii Practitioners with particular technical skills will be more positive about QA and measurement than others. For example, analysts will be more positive than programmers.

Summary of Findings

I present evidence addressing these hypotheses primarily in Chapters Six and Nine. I found convincing evidence to support hypotheses i, ii, iii and iv (with the proviso that I did not find that developers were resistant to QA and measurement, but they were less enthusiastic than managers). However, I could find no evidence to support hypotheses vi and vii. Hypothesis v proved very difficult to collect relevant data for.

4.2.2.3 Hypotheses about the implementation process

i QA and measurement initiatives are not usually properly resourced.
ii The goals of QA and measurement must be clear to practitioners if the programme is to succeed.
iii Effective QA and measurement must be transparent to practitioners if the programme is to succeed.
iv The usefulness of QA and measurement should be obvious to all practitioners if the programme is to succeed.
v Developers should participate in designing QA and measurement if the programme is to succeed.
vi QA and measurement initiatives are likely to be most successful when implemented in a bottom-up, incremental way.

vii Practitioners should have confidence in the integrity of measurement data if the programme is to succeed.

viii Feedback on QA and measurement needs to be provided if the programme is to be successful.

ix Any perceptions that data is being used inappropriately will make practitioners negative about measurement.

x QA standards are only marketing aids which do not necessarily improve software quality.

Summary of Findings

Because these hypotheses are fundamental to this thesis they are addressed throughout Chapters Five to Nine, but particularly in Chapter Eight. I found strong evidence in support of hypotheses iii, iv, v, vii and ix. However, I found very weak evidence to support ii and viii, while I found collecting evidence for hypotheses i and x difficult.

At the end of each Chapter in which substantive study results are presented (Chapters Six, Seven, Eight and Nine), I summarise how those results relate to the hypotheses. Furthermore, in the Conclusions Chapter I re-visit each hypothesis and assess the evidence for each and indicate where that evidence is located in the thesis.

5. Thesis organisation

This thesis is organised as follows:

Chapter Two: Study methods

In this chapter I outline the five phases of my study in which I collected data from twenty-six different companies. The purpose of the data was to draw an accurate and scientifically valid picture of the real state of QA and measurement practice in the UK software industry. I describe in detail my complete research lifecycle, the mechanics of which include using interviews, questionnaires and the Repertory Grid Technique. I also comment on the effectiveness of my data collection and analysis techniques.

Chapter Three: The currently reported state-of-the-art

In this chapter I discuss the current QA and measurement state-of-the-art. I add validity to my definition of this state-of-the-art, by including the views of various promoters and gurus that I have consulted during this research. By the end of Chapter Three I:

1. Identify the QA and measurement practices widely believed to be state-of-the-art.
2. Demonstrate the basis on which these practices are regarded as state-of-the-art.

Chapter Four: Reported best practice and recommendations

In this chapter I present the results of an extensive review of the company case studies published in the literature which describe the experiences of individual companies' implementing QA and measurement. By reviewing these published accounts I present the best practice in terms of the implementation of QA and measurement as it is reported in the software industry. I report in subsequent chapters how I use this as a basis for directly observing QA and measurement in companies, thereby testing the actual state-of-practice and comparing this reality with the state-of-the-art.

Chapter Five: An overview of real QA practice in industry

This chapter presents the results from my preliminary piece of research designed to provide a broad overview of the real current state of industrial practice. In this chapter I describe how I collected data from sixteen different companies with the aim of identifying those aspects of QA and measurement of most direct relevance to them. The results I present in this chapter provide the basis on which I subsequently designed more detailed and focused company case studies.

Chapter Six: Practitioners, QA and measurement: an industry-wide perspective

In this chapter I present the first tranche of results from the substantive part of my case study research. I provide a general analysis of the QA and measurement practices I uncovered during detailed case studies in four companies. In this part of the research I describe the experiences of nearly two hundred grassroots practitioners. I identify areas of commonality between the companies, and in doing so highlight aspects of QA and measurement that are likely to have industry-wide significance.

Chapter Seven: Practitioners and QA: company case studies

In Chapter Seven I focus further on two of my case study companies. In particular I discuss the QA practices I found when I conducted a more detailed analysis of these two companies. I describe how I followed QA development in these two companies over an eighteen month period, and discuss the company context of QA within each company.

Chapter Eight: Practitioners and measurement: company case studies

I follow the previous chapter with a chapter in a similar vein. However, here I pay particular attention to measurement in the same two case study companies.

Chapter Nine: Managers' and developers' experiences of quality

In this chapter I describe how I used the Repertory Grid Technique to gain a deeper understanding of practitioners' experiences of QA and measurement. I describe how I discussed QA with groups of managers and developers in another five companies. I also present qualitative data that allows us to understand the reality of what it means to develop software using QA and measurement. I report many frank views and comments from grassroots practitioners on the subject of developing software using QA.

Chapter Ten: Summary recommendations for successful quality assurance and measurement

In this chapter I bring together my analysis of theory and practice in guidelines for QA and measurement. These guidelines propose practices for the implementation and management of QA and measurement that have emerged as valid and practicable best practice.

Various pieces of advice are presented, including practical and valid advice for individual companies and suggestions for certification bodies and other more generally interested parties. By using these guidelines I believe companies will find that implementing effective QA and measurement is that much more attainable.

Chapter Eleven: Conclusions

In the final chapter I summarise my main findings on the current state of QA and measurement practice in the UK software industry. I discuss the gap between the state-of-the-art and practice across the industry, and I discuss the gap between official and actual practice within individual companies. I also re-examine the hypotheses that initiated the research and comment in detail on which of these are supported by my findings.

Chapter Two

Study Methods

1. Introduction

This chapter shows how I carried out this research. The particular data collection and analysis methods that I used are all well established and comprehensively covered in the literature. The methods used were questionnaires, interviews and the Repertory Grid Technique [Fransella & Bannister 1977]. I discuss both the theory and my application of these methods in this chapter.

The research methods used throughout this study are not often applied in software engineering research. They are more usually found in social science research. This is because they generate both quantitative *and* qualitative data. Historically, computing researchers have avoided collecting subjective and attitudinal data in favour of collecting 'hard' objective data. However, I found that so-called 'softer' research methods yield as much worthwhile data in software engineering environments as they are reported to do in social science environments [Bryman & Cramer 1994]. In particular the Repertory Grid Technique proved to be a very effective data gathering technique.

Throughout this research I collected data directly from practitioners. I have concerned myself almost exclusively with practitioners' experiences and perceptions of what their organisation was doing in terms of QA and measurement. I believe that practitioners, in contrast to senior managers, are the only people who know what the real QA and measurement picture is in a company. The situation practitioners describe may be very different to the 'official' QA and measurement situation. Consequently, when practitioners say, for example, that their organisation is not performing Function Point Analysis [Albrecht 1983], I have not verified that fact directly, I simply then asked managers what *they* understood to be the case. I found that the two views were frequently at odds with each other. Furthermore, if practitioners perceptions are inaccurate then this means that there has been a serious communication failure within a company's QA and measurement system.

In this chapter I concentrate entirely on describing the study methods I used to collect data on the current state of QA and measurement practice in the UK software industry. I do not describe how I identified the current state-of-art, as the primary method that I used was a conventional academic literature review and analysis. I, therefore, do not describe this standard process. My state-of-the-art findings are presented and discussed in Chapter Three.

In this chapter I start, in Section Two, by describing the principles and theory behind the main study methods I used. I go on to explain my application of those methods in Section Three, where I also describe in detail the research lifecycle that I adopted. In Section Four I comment on the limitations and effectiveness of the methods that I applied.

2. The study instruments

In this section I present an overview of the theoretical foundations of the study methods I implemented throughout this study. Many of these methods have been established for many years. This is reflected in the age of some of the texts that I used in researching my study methods, for example, one of the most useful texts I used was over thirty years old [Berdie & Anderson 1974].

2.1 Measurement

Measurement is clearly of fundamental importance in this research. I have adopted the following definition of measurement whether applied to measuring practitioner attributes or software attributes:

> *"[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, page 17]

Wherever possible I have attempted to remain true to the principles of measurement theory as described in [Fenton 1991].

2.2 Target populations - sampling

Identifying a representative target population or sample for research is fundamental to collecting data that is valid and useful. However valid the data collection instruments you use, data that are collected from an unrepresentative sample is not generisable to the population from which the sample was derived. The results will be skewed, and biased data that is not useful or applicable to anyone will be collected.

Unfortunately, identifying a sample that adequately reflects a specific population is not straightforward. Furthermore, even once an accurate sample has been identified, accessing and targeting that sample, *and* then actually getting responses from those targets is an equally difficult task. Moreover, non-response from a representative target sample seriously undermines representativeness (as there is likely to be unknown commonality between non-respondents).

Consequently, many studies suffer from data having been derived from samples of dubious representational validity [Easterby-Smith *et al* 1991]. Indeed this has been an historical problem in research, as particular target groups are always more difficult than others to persuade into participating. As a result of this [Easterby-Smith *et al* 1991] believes that "powerless" people (students, the poor and elderly etc.) are often over represented in research, whereas "powerful" people (like managers) often find ways to exclude themselves from research.

There are two main methods of sampling in common use. These are:

2.2.1 Census sampling

This is where the whole research population is targeted. For example, if the aim of the research was to measure morale in a specific company, then all employees in that company would be targeted.

The practicality of census sampling depends on various attributes of the study population - population size is, of course, most significant. However, where census sampling is practically and financially viable, implementing it avoids all of the representational issues (except those related to response rates) associated with other forms of sampling.

I was fortunate to be able to use census sampling on several occasions during this study.

2.2.2 Probability sampling

This is where a population will be represented by a specific proportion of the total population (the bigger the proportion the better). Using this sampling method each unit within a given population has a *specifyable* probability of being included in the sample [Bryman & Cramer 1994]. There are various ways of implementing probability sampling, the following are most common:

- **Random sampling**
  Each unit has an equal probability of being included in the sample. The random sampling method involves producing a complete list of all the units in a population (this is known as a *sampling frame*), and allotting unique numbers to those units. Numbers are then randomly generated (the number of which will be determined by the size of the sample proportion) and corresponding units are selected from the population for inclusion in the sample.

- **Systematic sampling**
  This is very similar to random sampling, except that once a sample size has been agreed (for example, 25% of the population will be included in the sample), each *n*th unit will be selected

from the sampling frame for inclusion in the sample (for example, a 25% sample will target each 4th unit).

In my study sampling was effectively twice as difficult. Not only did I have to identify a sample of companies to target from the software industry, but I then had to identify representative samples of practitioners from the population of practitioners within those companies that I targeted.

I discuss how I tackled sampling in Section Three of this chapter.

2.3 Piloting

The importance of piloting is always emphasised in the study methods literature. Great weight is put onto piloting every aspect of any study. In particular, [Oppenheim 1992] stresses the need to pilot all of the following:

— covering letters
— questionnaires
— methods of reducing non-response
— changes to any part of the study

In section 3 I discuss my piloting process.

2.4 Questionnaires

2.4.1 An overview

Although surveys are regularly used as an aid to social and political decision-making, it is a tool seldom used in the area of software engineering (though its use is increasing). Although there are known problems attached to using surveys, they remain valuable tools which can help to understand problems in any discipline. I used various types of questionnaires as a data collection mechanism during this research.

There are three basic types of questionnaire:

- **Mailed questionnaires**
  This is where a questionnaire is sent to respondents and completed by them within a specified time period. This kind of questionnaire is the easiest and cheapest to administer to very large samples, and has the potential advantage of eliciting very frank answers from respondents.

- **Self-administered questionnaires**

  This is where groups of respondents complete their questionnaires at a specified place and time, with a researcher on hand to answer queries.

- **Researcher-administered questionnaires**

  This is a questionnaire that is completed by the researcher in an interview-type situation. The researcher can explain any points of misunderstanding and probe further into particularly interesting answers. This type of questionnaire has the advantage that response rates are likely to be higher than mailed questionnaires and higher quality data can be collected. However, the boundary between a questionnaire and an interview begins to blur at this point.

2.4.2 Advantages of using questionnaires

Well constructed questionnaires can be a very good means of collecting data. The advantages of using questionnaires can include them being:

- **Effective and efficient**

  They are a cheap way of collecting a lot of data, potentially over a large geographical area. Analysing data from well designed questionnaires is also relatively cheap and easy.

- **Unbiased**

  A well constructed questionnaire can introduce a lot less bias that other methods of data collection. Unlike, say, interviews questions will always be presented in exactly the same way and everyone will be asked the same questions. Indeed questionnaires can provide a consistent and controlled data collection environment.

- **Highly participative**

  Contacting people and getting them to complete a questionnaire is also a lot easier than getting them to participate in other more sophisticated research methods. They are the easiest form of research for people to participate in. Most people are familiar with how to complete a questionnaire, and this encourages them to participate. Similarly, because questionnaires are such a regularly used study instrument, there is plenty of literature for researchers on the construction and application of effective and valid questionnaires.

- **Static**

  Although attitudes and experiences change over time, questionnaires are applied during a very short, specific timeframe. This means that the data collected can provide an accurate 'snapshot' of a situation at a given point in time. Other research methods usually take longer to apply and, therefore, are more susceptible to data pollution as a result of the environmental change that occurs during the lifetime of the data collection.

2.4.3 Disadvantages of using questionnaires

Questionnaires can have many potential problems, all of which need to be addressed in order for an effective and valid questionnaire to be constructed, these include:

- **Wording**

  Ambiguous wording is always an issue. It is very difficult to avoid misunderstanding and ambiguity in questions. Similarly it is often difficult to pitch the level of complexity at the right level, as it is difficult to word questionnaires so that they are not too detailed or technical, nor too vague or abstract. It is also necessary not to rely overly on memory in questions. Privacy should also be considered. Questions should not be too threatening or intimate, as respondents may become defensive in their answers or simply not respond.

- **Bias**

  Bias can be introduced if questions are leading, as questions can unintentionally reflect the researcher's own prejudices. This problem can be compounded by respondents' propensity to answer in a way that they think they should or think that the researcher wants. People also tend to only represent themselves positively in their answers.

- **Construction**

  The ordering of the questions may influence the way they are answered. Similarly the questionnaire itself may prompt respondents to think and, therefore, answer in new ways.

- **Responses**

  Poor response rates are often generated by questionnaires, and can produce a narrow range of responses. This leads to unrepresentative data being collected. Furthermore it may not even be clear who actually completed the questionnaire. Indeed, questionnaire 'fatigue' can be a particular problem for some populations where people are frequently targeted.

- **Data limitations**

  It is not usually possible with a questionnaire to probe respondents further or to ask for further explanation. This can make the data collected seem fairly superficial. The data may also suffer from lack of differentiation, as people are conservative in their responses to questions. This is usually because respondents are unsure about how any extreme answers will be interpreted and used.

These advantages and disadvantages are largely based on the discussions in [Oppenheim 1992], [Bryman 1990], [Berdie & Anderson 1974].

2.4.4 Designing questionnaires

Once a representative sample has been identified (the principles of this have already been discussed), it is crucial to achieve a high response rate within that sample. Questionnaire design, therefore, should both maximise responses and also elicit valid data.

2.4.4.1 Encouraging responses

There are many aspects of the design of a questionnaire that will make that questionnaire more attractive to complete and therefore more likely to achieve a higher response rate. These include:

- **A covering letter**

  Personalised and friendly covering letters are encouraging. They should offer a summary of study results to respondents, and provide some background about the study, without telling respondents too much about what the study is trying to ascertain (as this is likely to lead respondents). The letter should also tell respondents how long the questionnaire is likely to take to complete (shorter questionnaires are not necessarily more likely to be returned. Twenty minutes is, however, considered the maximum reasonable time.). Mentioning any study sponsors that the research has in the covering letter is also considered worthwhile.

- **Good aesthetics**

  Making the questionnaire aesthetically appealing, and easy to complete will also encourage responses. Coloured paper and pretty printing with lots of white space should all be considered. It should also look professional in terms of grammar, spelling, page/question numbering and printing etc. 'Tick box' questions are more likely to be answered than questions where the answer must be written. However, most importantly, the questionnaire must be *interesting* to respondents.

- **Clear instructions**

  Clear but brief instructions which explain how to complete the questionnaire should be included, and a return address must be clearly stated and positioned (some authors recommend putting the return address both at the beginning and end of the questionnaire). Including a stamped addressed envelope may also encourage responses (funding did not permit me to implement this suggestion).

- **Non-alienation**

  Avoid sub-sectioning questionnaires so that some sub-sections are only to be answered by specific types of respondents (i.e. avoiding questions which say *if you answered Yes to question 3 go to question 17...*). Do not assume much background knowledge from respondents as this will put respondents off who do not have that knowledge.

  Material inducements and entry to competitions may also be used to encourage responses, however, the effect of inducements is unclear, nor did I have the resources to implement inducements.

2.4.4.2 Collecting valid data

Designing questions to avoid bias and confusion is central to generating valid data from a questionnaire. As well as having to be mindful of the issues presented as questionnaire disadvantages, the following are some useful principles of constructing good questions:

Do:
- include a *Don't Know* option.
- provide mutually exclusive options that cover every eventuality.
- only ask one thing per question.
- include a balanced number of positive and negative questions.

Do not use:
- abbreviations.
- vague terms, like *most* or *several*.
- *ands* or *ors*.
- double negatives.
- jargon.
- use more than twenty-five words per question.
- an odd number of 'tick boxes' (as the middle one tends to be chosen).

This list is based on work presented in [Oppenheim 1992], [Berdie & Anderson 1974].

Question relevance is also crucial to collecting useful data. Experts on research methods caution against the temptation to collect data that seems interesting, but has no direct relevance to the study. In particular it is important that an explicit link between the hypotheses of any study and the actual data collected is maintained [Bryman & Cramer 1994]. Indeed questions should be designed which implement *dependent* and *independent* variables that model the study hypothesis (this is discussed more fully later in this section). It is common to find an opaque relationship between a study's hypotheses and the data that is actually collected during the study [Bryman & Cramer 1994].

To identify clearly the link between data collected and the study aims, many authors recommend constructing the analysis tables before designing the questionnaire. The data that needs to be collected is then clear and the questionnaire can be designed specifically to collect that data. However, I feel that there is a dilemma between only collecting data that is directly relevant to the aims of the study, and collecting data that becomes self-fulfilling and self-proving. I believe that other commentators have the same worries (Tom DeMarco voices such concerns about data in a slightly different context [DeMarco 1982]).

It is also sensible to design a questionnaire with ease of computer-based data analysis in mind.

There are also some well researched standard approaches to question design that can be implemented. One such approach that is frequently used is Lickert Scaling [Bryman 1990]. This is a standard multiple-choice answer format which is applied to questions where agreement to statements is measured. An example would be:

My Department...	Strongly Agree	Agree	Uncertain	Disagree	Strongly Disagree	Don't Know
1 produces quality software		✓				
2 reviews all code					✓	
3 has a quality culture	✓					

One advantage to adopting Lickert scales is that the data collected can reasonably be construed as being on an ordinal scale.

2.4.5 Generating questionnaire responses

Achieving a high questionnaire response rate is strongly related to how committed to participating in the research potential respondents are. One of the things that helps strengthen commitment, and therefore participation, is the commitment and involvement of a participating company's senior management. Senior managers must be seen to be committed to the research, and, even better, actively promoting participation in the research. Internal publicity for research is influential in securing high response rates.

On the other hand researchers must be careful not to *over* emphasise the involvement of managers, as this can be a negative factor. Participants must not think that the research is only a thinly veiled 'tool for managers' [Bryman 1990], as this will affect the way participants respond to the research. Unfortunately managers do sometimes try to monopolise and manipulate research for their own ends, and so researchers must tread a very careful line.

One way in which researchers can encourage commitment to their research is by confidentiality. The guarantee of strict confidentiality (not necessarily anonymity or non-attribution for follow-up purposes), is a very important factor in securing a high number of high quality responses. Similarly it is important to tell participants exactly how data is going to be used. Trust plays a crucial part in eliciting frank and honest answers from respondents. Indeed, Philippa Walker of Cockman Consultants and Partners says she is constantly surprised as to how honest and objective people are prepared to be:

> *"If you do a pay survey, many managers will say: 'You must be joking, they're all going to tell you they want more money', yet when you ask someone a question such as 'Do you think you're adequately paid to compensate you for your skills?', generally the answer will be a measured 'On balance, yes, I think I am'."*

2.4.6 Following-up questionnaire non-responses

It is emphasised throughout the literature that issuing follow-up prompts to non-respondents is a highly effective way to improve response rates. This can either be done by reference coding each questionnaire and then recording who has responded, or by issuing general reminders to everyone (regardless of the possibility of them having actually completed the questionnaire). It is also suggested that several such reminders are issued, and that on at least the first occasion another copy of the questionnaire be included. Phone call reminders are said to be the most effective form of follow-up [Berdie & Anderson 1974]. This process also provides researchers with a valuable opportunity to find out *why* people are not responding. As there may be a flaw in either the design, administration or targeting of the questionnaires that researchers may not be aware of.

2.4.7 Analysing questionnaire data

For the analysis phase of any research project to yield useful results all previous study phases must have been implemented well. Without these precursors *any* data analysis is pointless. The data analysis techniques that I implemented were *inferential statistics* [Bryman & Cramer 1994]†. This was because I was trying to discover the probability that the data I collected from my samples, were a true reflection of the population from which my sample had been taken.

† I relied heavily on [Bryman & Cramer 1994] during my data analysis research.

2.4.7.1 Causality

Data analysis tries to establish whether relationships that we suspect might exist, actually do exist. Hypotheses are tested to uncover causality. The following three factors must be considered when looking for causality [Bryman & Cramer 1994]:

1. That an apparent relationship exists. For example, that practitioners with previous QA experience seem to have more positive attitudes to QA.

2. That the apparent relationship in not spurious. For example, that the relationship is not just a quirk of the data or a result of some other factor, like those people being keenest about everything.

3. That cause precedes effect. For example, that people did not already feel more positive about QA before they had QA experience. This is often very difficult to prove, not least because related data is usually collected at the same time.

To analyse for potential relationships requires the questionnaire (or whatever study instrument is being used) to implicitly embody two things:

- **The null hypothesis**
  In all research researchers must begin with the null hypothesis, in that they assume that the relationships they are interested in do not exist. Only once the null hypothesis has been rejected can a relationship be shown to exist. I will discuss the basis on which the null hypothesis can be rejected later.

- **Independent and dependent variables**
  The questionnaire must contain questions that will collect data modelling the relationships the researcher is interested in. Questions must model independent and dependent variables. Independent variables are those variables that effect other variables, for example, gender or previous QA experience. Dependent variables are those that are effected by independent variables, for example, height or attitudes to QA. If a particular independent variable can be shown to affect a dependent variable then that effect may be controllable.

Furthermore, data analysis must ensure *internal validity*. This means that there must be no other plausible explanation for any relationship that has been identified. Internal validity is very difficult to prove using surveys and for this reason they are known as *correlational studies*. Conventional scientific experimentation addresses internal validity more directly by using control groups and placebos.

2.4.7.2 Types of questionnaire data

The type of measurement data collected has a significant impact on how data can be analysed and which statistical techniques are appropriate to apply. This is often overlooked in statistical analysis and inappropriate analysis methods are applied to data. It is important that each piece of data collected is categorised according to its type, and then only appropriate analysis performed on that data (for example, it is not appropriate to calculate the mean of some types of data, for example, nominal data).

The types of data that exist are:

- **Nominal**
  This is classification data only. In this study nominal data was the most common type of data collected, for example, job title.

- **Ordinal**
  This data type accounts for basic ordering of nominal data. Attitudinal data is often ordinal. Ordinal data was also collected in this study, for example, the accuracy of metrics data.

- **Interval and ratio**
  There are meaningful gaps between the units in this measurement scale, furthermore, ratio data has a zero point. A small amount of this kind of data was collected in this study, for example, time spent on QA activity.

2.4.7.3 Statistical analysis methods

There are three basic types of statistical analysis. These are:

- **Univeriate analysis**
  Single variables are analysed here and relationships between variables are not explored. Frequency tables are the usual implementation of such analysis. This form of analysis is used extensively in this study.

- **Bivariate analysis**

  Potential relationships between two variables are examined using this type of analysis. The data is analysed to see whether differences in the distribution of two variables are statistically significant and not just there by chance. A classic application of bivariate analysis is comparing the independent variable gender to a dependent variable, say, height.

  The simplest form of bivariate analysis is via crosstabulation. Again, this form of analysis is used extensively in this study. More details on performing bivariate analysis via crosstabulation is given later.

- **Multivariate analysis**

  Connections between groups of variables are explored using these types of statistical tests. This is the most complicated set of statistical techniques and probably reflects the complex relationships that exist in the real world. I did not use this class of statistical technique in my study.

2.4.7.4 Establishing relationships

The data collected during this study were mostly nominal and from unpaired groups (i.e. there was no existing relationship between the two groups). In addition, I was interested only in two dimensional relationships between variables and those variables were mostly coded alphabetically. Therefore, I concern myself with discussing only the small range of statistical tests that are appropriate in the circumstances described, i.e. bivariate, non-parametric tests between dependent and independent variables in order to reject or accept a null hypothesis. In fact the limitations of my data (and because I wanted to keep my analysis simple and transparent) means that it is only necessary to discuss the details of: frequency tables, crosstabulation, the chi-square statistic and Pearson's significance level. The statistical analysis tool SPSS [Frude 1993] calculates all of these statistics and so I do not describe the mathematical formulas that are automatically implemented by SPSS. Appendix Four provides some sample SPSS output.

- **Crosstabulation**

  Crosstabulation is a very simple and transparent implementation of bivariate analysis. Furthermore, potential relationships can quickly be identified for more detailed follow-up. Crosstabulations are implemented via contingency tables, as shown in Table 2.1.

  Table 2.1 shows the results of crosstabulating the answers to two questions. The questions ask how often practitioners are involved in QA and how long their service with a company is. The crosstabulation explores whether or not there is likely to be a relationship between the two variables. Each cell in the table contains four numbers:

1. The raw number.
2. The row percentage.
3. The column percentage.
4. The table percentage.

So, for example, in Table 2.1 the numbers in the first cell show that: two practitioners who have worked for a company for less than one year are *never* involved in QA; 8.7% of practitioners who are *never* involved in QA have worked for companies for less than one year; 40% of practitioners who have worked for companies for less than one year are *never* involved in QA; 1.1% of all practitioners have worked for companies for less than one year and are *never* involved in QA.

Q7VII QA by Q1 Service Length

Q1 — Page 1 of 1

Q7VII	Count / Row Pct / Col Pct / Tot Pct	Up to 1 Year a	1 to 5 Years b	5 to 10 Years c	Over 10 Years d	Row Total
Never	a	2	8	8	5	23
		8.7	34.8	34.8	21.7	13.2
		40.0	25.8	12.9	6.6	
		1.1	4.6	4.6	2.9	
Occasionally	b	2	17	36	36	91
		2.2	18.7	39.6	39.6	52.3
		40.0	54.8	58.1	47.4	
		1.1	9.8	20.7	20.7	
Frequently	c	1	6	18	35	60
		1.7	10.0	30.0	58.3	34.5
		20.0	19.4	29.0	46.1	
		.6	3.4	10.3	20.1	
	Column Total	5	31	62	76	174
		2.9	17.8	35.6	43.7	100.0

Table 2.1: A standard contingency table

It is important that column and row percentages are considered when contingency tables are drawn up. Calculating column or row percentages affects how the data is viewed, and will be dictated by what aspect of the data needs to be highlighted. However, it is usual for column percentages to be implemented when the independent variable is the horizontal variable (though identifying dependent and independent variables is not always that easy, as determining which factor came first is sometimes difficult to establish). In Table 2.1 service level is the independent variable and QA involvement the dependent variable.

- **The Chi-square test**

Once a relationship between a dependent and independent variable is suspected and represented in a contingency table, this relationship can be explored further using the chi-square test. This test calculates both the expected and actual frequency of cells in a contingency table. The test calculates the expected frequency on the basis of the frequency that would occur if no relationship existed between the variables and the data was randomly distributed (by default SPSS does not show expected cell values). A probability is then calculated that indicates the likelihood that the observed relationship is a real relationship and not there by chance or by sampling error. The greater the difference between observed and expected frequencies the higher the chi-square value is. For example, the data in Table 2.1 produces the chi-square value shown in Table 2.2. This example has an acceptable Pearson's significance level.

```
Chi-square                     Value          DF              Significance
--------------------        -----------      ----            ------------

Pearson                      15.74470          6                 .01519
Likelihood Ratio             14.74954          6                 .02230

Minimum Expected Frequency -      .661
Cells with Expected Frequency < 5 -     4 OF    12 ( 33.3%)
```

Table 2.2: A chi-square significance table

The larger the contingency table the more likelihood there is of recording a higher chi-square value. To normalise this *degrees of freedom* are factored into the chi-square calculation (SPSS does this automatically).

- **Pearson's significance levels**

In order to finally decide whether to reject the null hypothesis and accept that a relationship actually exists, we must decide on the risk that we are prepared to take on that relationship existing only in the sample and not in the population. Standard acceptable risks include probabilities of: 0.05, 0.01, or 0.001. A probability of 0.05 means that there is a 1 in 20 chance of rejecting the null hypothesis when it should be accepted, and is generally considered to be the highest acceptable risk. Obviously 0.01 (a 1 in 100 chance) and 0.001 (a 1 in 1000 chance) are more stringent risk criteria. However, these measures only relate to the *risk* of wrongly rejecting the null hypothesis and do not indicate the strength of any relationship. Indeed, recording a 0.0001 value only makes it highly likely that there is a relationship between the variables, it says nothing about how strongly those variables are related.

The characteristics of the data and my use of the chi-square statistic means that Pearson's significance level was the most appropriate statistic for determining risk in this case.

2.4.7.5 The limitations of statistical analysis methods

Although the particular statistical techniques that I describe are highly suitable for this study, they have the following limitations:

- **Proof**

  They can never be used to actually prove a relationship exists. The best they can do is either disprove a relationship or offer collaborative evidence of one.

- **Strength of relationship**

  No indication of the strength of relationships is possible (this can be done using correlation analysis, although this is not very reliable with nominal data).

- **Reliability**

  They are not reliable when more than 20% of cells have an expected frequency of less than five (SPSS generates an automatic warning if this is the case).

- **Data type**

  They are most useful for the analysis of nominal data.

- **Table size**

  Dubiously high results are generated when applied to a 2x2 crosstabulation (though SPSS automatically calculates the Yates statistic to overcome this limitation).

2.4.8 Reporting questionnaire results to participants

Results can be produced which summarise the data collected within one company and that compare several participating companies. Promising and providing such summaries help avoid the following potential problems:

- Initially inducing suitable companies to participate in the research.
- Eliciting responses from respondents in participating companies.

Senior managers are, understandably, reluctant to commit company time and resources to participating in research projects, as they usually cannot see a direct payback for doing so. Managers also know that any data that is collected during research has value, and without any 'payment' for it many are reluctant to allow access. Consequently a summary of results is often a tempting inducement for company managers to participate, especially if it means they get a limited amount of data on other companies (however, they are never as keen for their results to be shown to other companies). This kind of inducement also works well for individual practitioners within companies. They too are interested in how their experiences compare to their colleagues. Indeed, [Marsh 1982], suggests that it is particularly effective to issue both types of progress reports throughout the study, rather than just one final summary.

Unfortunately producing these summaries serves no useful purpose for the research, and so there is no real payback for researchers [Bryman 1990].

2.5 Conducting interviews

Conducting high quality interviews that generate valid data is a difficult and skilful task. However when implemented effectively interviews can produce very substantial data. Burgess describes interviews thus:

> *"... the opportunity for the researcher to probe deeply to uncover new clues, open up new dimensions of a problem and to secure vivid, accurate inclusive accounts that are based on personal experience."* [Burgess 1991]

There are many issues to consider when designing interviews, the biggest being the avoidance of bias. It is easy for an interviewer to lead interviewees and so not only must questions be non-leading, but an interviewer's tone must also not lead. Care must also be taken that interviewees are not tempted to tell the interviewer what they think they want to hear. Furthermore, not everything will be noted during an interview and this will also introduce selectivity into the data collected.

2.5.1 Types of interviews

- **Free flow interviews**

  Unstructured, free flowing, open interviews are most suitable when interviewing 'key informants'. [Oppenheim 1992] believes that conducting such interviews are an effective way to uncover key issues. However, [Easterby-Smith *et al* 1991] is much less enthusiastic about this style of interviewing, asserting that they:

  > *"... are not likely to produce a clear picture of anything. Nor are they likely to answer any questions or highlight issues worth following up."*

- **Structured and semi-structured interviews**

  These are the most usual format of interviews and are based around a specific set of questions. These questions are delivered to interviewees in the same way, in the same order etc.

2.5.2 Interviewing techniques

Various authors present advice for conducting effective interviews, for example [McCrossan 1985], including:

- Do not re-order the wording or delivery of questions.

- Ensure that the interviewee has:
  - understood the question
  - answered the question clearly
  - only supplied one answer to the question

- Use only non-biased and non-leading terms, for example *"Can you explain that a little more fully?"* *"How do you mean?"* *"In what way?"*

  As opposed to saying things like *"Do you mean that you think..." etc.*

2.5.3 Recording interviews

How interviewers record interviews is an important and controversial issue. An interviewer can either take notes or use an audio recorder. Both of these methods have advantages and disadvantages, the most important of which is the fact that although audio recording allows the interviewer to concentrate only on listening and talking rather than writing, and produces a complete recording of the interview, audio recording can make people more defensive about what they say. Note taking, on the other hand, tends not to worry people as much, but can have the disadvantage of distracting the interviewer from the interview and only selectively representing the interview.

2.6 The Repertory Grid Technique

The Repertory Grid Technique (RGT) [Fransella & Bannister 1977] is based around understanding and changing people's perceptions of specific stimuli. It is a qualitative research and analysis method that is based on Kelly's *Personal Construct Theory* [Kelly 1955]. Originally the technique was developed for use in psycho-analysis, but has been adapted for more general use. One of my hypotheses is that practitioners' perceptions of QA and measurement play an important role in the success or failure of any programme. I, therefore, used this technique to try and understand what practitioner perceptions really were, and how programmes could be changed to maximise practitioners' positive perceptions and minimise their negative perceptions.

The aim of the technique is to, first, understand personal perceptions of a particular issue and then to identify how people have constructed those perceptions. The idea is that not only will the method actually uncover people's perceptions, but how those perceptions are constructed is also uncovered. Once those constructions are understood changing some of the negative constructions that have led to negative perceptions might be possible. When the method is implemented in its entirety detailed cognitive maps or grids can be produced (I did not apply the method to such an advanced level).

The method is applied in an environment where the researcher talks to RGT participants. I applied the RGT to eight separate small groups of developers and managers. During a RGT session the researcher classifies what participants say according to the following:

- **Elements**
  The objects of analysis. In my research these included software quality, development methods and processes, quality management, quality certification and so on.

- **Constructs or qualities**
  These are the descriptions of elements that emerge during the discussion. To apply the method at an advanced level researchers should elicit bi-polar descriptions of elements so that contrasting is possible. For example, in my sessions I asked developers to describe a manager with a good approach to quality and a manager with a bad approach to quality. I also asked each group of developers to describe how their views of quality differed from those of managers and how their views were similar (I also asked the managers the same question).

- **Links**
  This is how elements and constructs are linked to one another (at an advanced level these are represented in a mathematical matrix).

RGT is a method that seeks to get to the root of people's perceptions and enable some control over the driving forces behind those perceptions. It is necessary for researchers to steer and prompt RGT conversation to elicit the type of data that the method aims to collect. Bannister & Fransella suggest ways of eliciting this data via *laddering* and *pyramiding*:

- **Laddering**
  This attempts to elicit extra data on an element at a higher level of abstraction. Asking people *why* and *how* is the main way of doing this. For example, researchers can follow up any statement made by asking: *Why is that important?* or *How do you know (*for example, *when software is of high quality)?* Or simply asking them to repeat what they have just said (as they will usually do this in a slightly different way).

- **Pyramiding**

  This attempts to get more general data on a specific element. For example, during a description of a good quality manager, the researcher might broaden the debate in the following way:

Question:	*What kind of person is manager x?*
Possible answer:	*She is flexible.*
Question:	*What kind of manager is flexible?*
Possible answer:	*A manager who is willing to listen to the developers.*
Question:	*What kind of manager is not willing to listen to the developers?*
	How do you know when a manager is not willing to listen?

Once positive and negative attributions have been identified using these methods it should then be possible to identify strategies to improve the things that lead people to have negative perceptions. It should also be possible to identify areas where prejudice and ignorance has led to negative perceptions and then to embark on attempts to address these. The positive attributions can be emphasised and should prove very effective as these are the things that are most important to people and are most likely to improve perceptions. I found the RGT was a particularly effective way to uncover what practitioners really think about quality issues, though was a difficult technique to apply in full.

3. Implementing the study methods: the phases of the study

This section describes how the study methods were implemented during this study. The section is divided into five separate phases.

3.1 Phase One: a pilot study

The results of this pilot study are presented and discussed in Appendix Two.

To test the effectiveness of my approach to questionnaires and interviews for collecting quantitative *and* qualitative data in a computing environment, I conducted a basic pilot study.

The specific aims of this pilot were to:

1. Evaluate the questionnaire.
2. Test my administration of the questionnaire.
3. Test my approach to interviewing.

4. Elicit feedback from participants on the methodology and content of the study.
5. Determine the relevance of the data I was collecting.
6. Collect data that accurately represented the pilot department.

3.1.1 The pilot department

The pilot study was applied to the Computing Services department of a university. I suspect that this Department is probably typical of many Computing Services Departments within companies whose main business is not computing. The Department has the following features:

- It has sixty-three staff in total.
- It develops and supports many different applications.
- It supports several thousand users.
- Many different languages, development tools and methodologies are in use across the Department.
- The software development process is very immature.
- Its managers are keen to deliver high quality products and services.
- There are almost no formal procedures in place across the Department.
- Pockets of staff have their own semi-formal ways of doing things.
- No measurement programme was in place.

However, managers were already looking to implement practicable standardised approaches to software development, maintenance and support across the Department. Managers were particularly interested in more formal and effective ways of operating the Department as its systems were to be quality audited.

3.1.2 The pilot questionnaire

In designing the pilot questionnaire I followed the questionnaire design principles that I described in Section Two. Particular design points to note include my implementation of:

- dependent and independent variables
- Lickert scales
- nominal and ordinal measurement

I also customised the pilot questionnaire to the particular pilot department and was, therefore, able to be very precise about which work groups I was interested in. I did not customise subsequent questionnaires in this way and people tended to get confused about whether they should reveal what their organisation, department, team or other work group was doing in terms of quality assurance and measurement.

The pilot questionnaire can be found in Appendix Three.

3.1.3 The pilot survey

I distributed questionnaires to the thirty-four staff in the Department who were involved in software development or the management of software development. I excluded staff who did other things (such as user support, hardware support, administration etc.). However, in effect I was able to census sample.

Twenty-three completed questionnaires were returned, which meant that I had an extremely high response rate of 68%. An average response rate of between 30-40% is more usual [Oppenheim 1992].

Although the response rate for the questionnaire was good, the target population remains very small. This means that the results from this pilot study have poor Pearson's significance levels and cannot be generalised. However, the high response rate does mean that my results accurately characterise this particular Department.

3.1.4 The pilot lifecycle

I applied the whole study lifecycle during this pilot*, so that I could identify methodology weaknesses throughout the lifecycle. I followed the best practice study lifecycle principles that I describe in Section Two and as promoted by [McNeill 1990]. My pilot study lifecycle included:

1. Negotiating with managers about study participation, logistics and administration.
2. Publicising the study within the Department.
3. Conducting a free-flow interview with the Head of Department.
4. Designing the questionnaire.
5. Applying the questionnaire.
6. Conducting follow-up structured interviews.
7. Analysing data using SPSS.
8. Drawing conclusions from the results.
9. Feeding back and discussing results with managers.

* I did not pilot the Repertory Grid Technique at this stage.

3.1.5 Evaluating the pilot

Despite adopting a rigorous approach to data collection, I fell into several of the traps I warn of in Section Two.

3.1.5.1 Questionnaire evaluation

During this pilot study I found questionnaires were an effective and reasonably efficient method of collecting data. However, I did find, as [Berdie & Anderson 1974] suggest, that questionnaires were less efficient on smaller samples (such as the sample size in this pilot study). I also found that when I analysed the questionnaires, and looked at the feedback I received from pilot study participants, the pilot questionnaire (Appendix Three) needed improvement. These improvements resulted in the questionnaire in Appendix Three. The specific improvements I made to the pilot questionnaire included:

- **Aesthetics**
  Improving its aesthetics so as to make it more appealing to respondents.

- **Re-focusing**
  Only including questions of direct relevance to the study (for example, removing question nine.).

- **Differentiation**
  Removing or improving questions which yielded bland or insufficiently differentiated data (for example, improving question twelve). Also re-considering questions with a wide spectrum of multi-choice answers, as respondents tended to be conservative and choose mid-point answers rather than extremes. I addressed this by removing Lickert scale questions (though this did not prove to be an improvement).

- **Measurement**
  Including more questions specifically relating to measurement.

- **Open questions**
  Reducing the number of open questions (for example, re-stating questions seventeen and eighteen).

- **Ambiguity**
  Deleting ambiguity by, for example, removing double negatives and 'double-barrelled' questions (for example, re-stating question eleven).

- **Coding**

  In the pilot I coded each question for personalised follow-ups, but people were very worried about how I would really use attributable data. I, therefore, removed personalised codes from subsequent questionnaires.

- **Feedback**

  I introduced a new section inviting participants to evaluate and comment on the questionnaire.

I also hoped that by improving my main study instrument in this way I would elicit less *Don't Know* answers. The proportion of such answers was far higher than I expected, and was associated with questions that I could reasonably expect respondents to know about (for example, question 15iii asked whether the Department used a quality manual and had a 50% *Don't Know* response rate). [Berdie & Anderson 1974] found that people are generally reluctant to admit that they don't know things and so I was forced to conclude that many of my *Don't Know* answers were a function of poorly worded questions.

I found that my process of administering the questionnaire worked well and I implemented no major changes to this.

3.1.5.2 Interview evaluation

During the pilot study I also discovered that it was easy to conduct too many follow-up interviews. These interviews turned out to be:

- **Time consuming**

  Each took about two hours, and even in this small pilot I needed to conduct three such interviews. Given resource and time constraints, this would be untenable on a larger scale.

- **Difficult to keep focused**

  Interviewees frequently got side-tracked and this made interviews ineffective and inefficient. As an interviewer I found controlling interviewees quite difficult. This was especially the case when I interviewed managers. Managers seemed much more used to this kind of situation and were adept at controlling and manipulating the agenda ([Easterby-Smith *et al* 1991] warn of this problem).

- **Unproductive**

  They yielded too much duplicate data.

- **Hard to record accurately**
  I tried both note-taking and audio recordings. Both had advantages and disadvantages, but I decided to restrict myself to only note-taking in subsequent study phases. I found audio recording disruptive to the interview and entailed a lot of transcription work. I also found that the couple of minutes that interviewers took to write down comments made by interviewees acted as very subtle prompts, and gave interviewees opportunities to add extra, free flow, comments. However I was most concerned about the effect on practitioners' answers when they were recorded. I felt that practitioners were more candid when only notes were taken.

I, therefore, decided to confine interviews to a small number of pre-questionnaire 'fact finding' interviews, and only conduct follow-up interviews if I needed to clarify specific issues. Although I consulted the literature on interviewing techniques [McCrossan 1985], I realised that gaining enough skill to conduct effective interviews, proved more impractical than I anticipated (given time and resources constraints). I do, however, believe that given more practice, and, possibly training, interviews would have yielded a great deal of valuable data.

I have shown in this section that there were a lot of improvements to be made to my initial approach, and indeed, I implemented all of these improvements in subsequent phases of my research. However because so many weaknesses emerged during this phase I did 'go back to the drawing board' and back to the literature and thoroughly re-think. I then effectively re-piloted the questionnaire in the next study phase.

3.2 Phase Two: uncovering the issues

The results of this phase of the study are presented and discussed in Chapter Five.

3.2.1 Aims of the phase

During this phase of the study I aimed to identify the QA and measurement issues that were most important to a broad range of software companies. I specifically sought breadth of company coverage so that I would be able to identify a reasonably representative sample of current issues. At this point I decided to limit my research to companies which were already implementing QA and measurement, so that I could explore relevant experiences in practice. I achieved this by distributing a questionnaire (Appendix Three) at a software measurement symposium [IEEE 1994]. The questionnaire I used was a customised and improved version of the questionnaire that I used during the pilot study. I followed this questionnaire up with informal discussions with a few respondents. Each respondent effectively represented their organisation - this research method is known as Key Informant Research [Bryman 1990].

3.2.2 Response rate

Although I issued the questionnaire to all ninety-two Symposium delegates, the completed questionnaires that I received from the academic delegates proved not to be relevant, and were discarded. I received sixteen completed questionnaires from industry delegates. The final response rate for the survey (of industry delegates only) was 23%. Although this may seem a poor response rate (and I was certainly disappointed with it), it is in fact not too far below average for a survey of this nature [Oppenheim 1992].

The respondent profile tended to be quite senior software engineering staff, most of whom were either managers or QA practitioners. The self-selecting nature of this survey also meant that, on the whole only delegates with first-hand knowledge and a good understanding of their company's quality programme responded. All sixteen companies also seemed to have a very serious approach to quality.

Nevertheless, some of the reported problems associated with surveys are very relevant to this study. For example, representation: the target population (i.e. the industry contingent at the Symposium) by participating in this event are not likely to be truly representative of the whole of the software industry. Furthermore, a 23% response rate makes it difficult to claim that the respondents were even representative of the Symposium's industrial contingent. It is also practically impossible to generate statistically significant results from such a small respondent population, and consequently many of the results in this study are not significant results in a statistical sense.

3.2.3 Evaluation of the phase

This part of the study served primarily as a 'reconnaissance' of the quality issues that were of most concern to companies, and in the process I gained a useful in sight into QA and measurement in sixteen different companies. I used the results from this phase simply to identify promising areas of QA and measurement to look at in more detail in later phases of the study. In addition, the revised and improved questionnaires and approach to interviews were re-piloted during this phase. So although the data gathered during this phase were disappointing, the broad aims of the phase were fulfilled.

3.3 Phase Three: a study of industry

The results from this analysis are reported and discussed in Chapter Six.

I did the first part of my substantive research into the current state-of-practice in the UK software industry in this phase of the study. Over a six month period I analysed the QA and measurement practices of four companies. The objective of this part of the study was to conduct detailed research into QA and measurement in a small number of companies. I also began to follow-up the QA and measurement issues that I identified in the previous phase of the study. I collected a mixture of quantitative and qualitative data during this phase of the study.

I looked at a variety of different types of companies to try and identify commonality in the way that companies were implementing and managing QA and measurement. I did this by:

1. Identifying the organisational and management infrastructure of companies.
2. Generating a picture of how companies were tackling QA and measurement.

The way I implemented this phase of the study is as follows.

3.3.1 Targeting companies

All four companies in this phase of the study had well established QA and measurement procedures. I identified these companies as suitable potential participants by using public domain information (this was usually by using information from published articles and papers). Using this method I identified twelve potential participant companies, and I contacted the quality manager at each of them. Four of the twelve companies agreed to participate in the research.

Whilst all four companies satisfied my criteria of having both QA and measurement in place, the maturity of the QA and measurement processes varied between the companies. However, none had an optimised QA or measurement process. Table 2.2 presents a brief overview of the four companies involved in this phase of the study.

Company Pseudonym	Size (Employees)	Type	Software Developed
Information Systems (IS)	Over 10,000	Privatised public corporation	Information systems
Embedded Systems (ES)	Over 10,000	Engineering company	Safety critical
Technical Systems (TS)	About 5,000	Technical equipment manufacturer	A range of applications
Software Systems (SS)	50 - 100	Software house	Information systems

Table 2.2: The companies in the study

3.3.2 Fact finding

For each company that agreed to participate in the study I conducted a small number of fact finding interviews. I interviewed managers who had a high level of corporate responsibility for QA and/or measurement. Each interview lasted about two hours during which time I identified the context, framework and content of the official programmes. At this point I wanted to establish what practices the company believed were currently practiced in its programme. My aim was to compare this 'official' view of QA and measurement with what practitioners subsequently said was actually happening. I also gathered basic demographic information about the four companies. The information I collected was based around the proforma in Appendix Three. In particular I collected data on the:

- Staff profile.
- Application profile.
- Development environment.
- QA activities.
- Measurement data collected and how it was used.
- Implementation and management of QA and measurement.

3.3.3 Applying questionnaires

I liaised with quality managers in each company to construct a structured sampling frame and to identify target groups of practitioners. The resulting target populations consisted of a cross-section of junior and senior developers and managers, all of whom had been exposed to QA and measurement in their company. The target population also included many different types of practitioners (analysts, programmers, testers, project managers, QA practitioners etc.). There was also a range of experience and service levels represented within each sample.

Company Pseudonym	Target Population	Questionnaires Issued	Returned	%
Information Systems	A random sample from the whole computing function	125	103	82
Embedded Systems	A census of one small SE team	24	20	83
Technical Systems	A random sample from one large SE project	63	35	56
Software Systems	A census of all SE staff	32	20	63
Total		244	178	73

Table 2.3: The companies in the study

Table 2.3 gives more details of each company's target population and shows how target populations were selected in each company. It shows that I carried out a consensus survey in some companies and semi-random sampling in others*. The reasons for this are discussed later.

The questionnaire I applied in each of the four companies can be found in Appendix Three. The specific aims of the questionnaire were to:

1. Identify which QA and measurement activities were taking place in the companies.
2. Find out how QA and measurement were initially implemented and subsequently managed.
3. Establish the contributions that individuals and groups of practitioners made to QA and measurement.
4. Solicit views and experiences of QA and measurement.
5. Identify the critical success factors associated with the introduction of QA and measurement.
6. Record practitioner attitudes to QA and measurement.

I adopted a best practice approach to administering questionnaires, as tested and improved via the pilot study, and as described in Section Two of this chapter. Specifically I:

1. Secured the firm support of company managers from the outset. These managers then publicised the study within their companies.
2. Allowed a specified, but generous length of time for questionnaires to be completed and returned.
3. Prompted practitioners who did not return their questionnaire within the stated deadline.
4. Promised and provided feedback to participants on the study results.

Using this method I achieved an overall response rate of 65%, with 182 questionnaires being returned in total. Table 2.3 shows how the response rate varied between the companies. This exceptionally high response rate is probably the result of the following circumstances:

- **Topicality**

  Although all the programmes in this study were maturing, none were very long established. This meant that the programmes were still current issues, and practitioners had a lot they wanted to say about them.

* I refer to it as semi-random sampling as I did structure the sampling frame so as to guarantee representation from all major sub-groups of practitioners (for example, testers, analysts etc.).

- **Desire for evaluation and improvement**
  The managers of the programmes fully and publicly supported this study. Furthermore there had been no previous formal evaluation of the programmes and the managers were, therefore, keen to encourage practitioners to contribute their views to this evaluation. Developers themselves also seemed keen to contribute, perhaps expecting their contribution to be used for programme improvement.

- **Independence of the study**
  Practitioners were confident that what they said was in confidence, and could not be used against them by their managers. As a result they probably felt more able to contribute in an open and honest way.

The response rate in TS was, however, very disappointing. I believe that this was because many practitioners employed by this company were contractors. I suspect that these practitioners had very little allegiance to TS and, therefore, had little incentive to contribute to the study. In addition the mobile nature of this group of staff made targeting effective follow-ups difficult.

Such a high response rate gives confidence that my results are an accurate representation of the programmes in most of the companies I studied. Although, like all case studies it would be wrong to generalise the results to the software industry as a whole [Kitchenham *et al* 1995], as it is difficult to prove that these companies are truly representative of the software industry. However, the relatively large number of responses that I collected and analysed in this phase of the study meant that I was able to produce highly statistically significant results. Consequently, I believe the results presented here give many interesting insights that will be relevant and applicable to other companies.

3.3.4 Analysing the results

I analysed the data from completed questionnaires using the statistical analysis package SPSS. Using this package I was able to analyse the data I collected extensively. I concentrated particularly on producing:

1. Frequency tables
2. Crosstabulations
3. Chi-Square tests
4. Pearsons correlation significance levels

Sample output from this analysis can be found in Appendix Four.

Customised feedback, on the basis of these results, was also provided to each participating company.

3.3.5 Evaluating the phase

In this phase of the study I decided not to personally encode questionnaires (i.e. put unique identifiers on individual's questionnaires), but make them completely anonymous. About four weeks after issuing the questionnaires I then sent reminders for returned questionnaires to everyone that the questionnaire was initially targeted at. This meant that those people who had already returned their questionnaires also got prompted. I reasoned that this was better than respondents thinking that a reference code on the questionnaire was really to relate responses to people, rather than just to identify non-respondents. I made this decision as I had coded the pilot questionnaires, and many concerns had been voiced from pilot participants. With hindsight I believe that this was the wrong approach. Had I *told* respondents what the reference code was for at the outset, I would then have had the advantages of:

- Encouraging more unprompted responses (because people knew from the outset that I was going to pester them for a response).
- Only issuing reminders to those people who had not responded.
- Collected valuable data about the non-respondent group and whether there was anything significant about them.
- Been able to ask specific people for the reasons why they had not responded.

In one of the companies I had a problem with senior managers manipulating the research for their own ends ([Bryman 1990] warns of this possibility). Very early on managers in IS insisted on sending questionnaires to practitioners themselves via their internal mail (a sensible suggestion I thought at the time). However, when completed questionnaires were returned (via a central IS collection point) all of the questionnaire covers were missing. I subsequently discovered that new covers had been put onto the questionnaires and distributed under the auspices of an internal evaluation. Clearly I cannot be certain that all questionnaires were then returned to me, nor can I be certain that respondents were not influenced by the questionnaire purporting to be internal.

With hindsight I believe it would have been more valuable to implement self-administered questionnaires, rather than the mailed questionnaires. I probably had ideal, self-contained samples to make this method effective. This probably would have overcome any misunderstandings and possibly also generated a higher response rate. Whether companies would have agreed to this form of administration is, however, debatable.

3.4 Phase Four: detailed company case studies

In Chapters Seven and Eight I present and discuss the results of this phase of the study.

After having completed an overall analysis of the issues in the four companies (as discussed above), I decided to focus further on the two companies with the longest established programmes. These two companies had some key similarities, both companies:

- Are very large and long established with sophisticated internal organisations.
- Have a large software development function (each with over 400 software staff).
- Are dependent on in-house software systems to support or enhance their main (non-software) products.
- Use many new and innovative development methods and tools.
- Have maturing QA programmes.
- Have had a measurement programme for at least two years.

Characteristic	Embedded Systems	Information Systems
General profile	• Very large engineering company	• Large information processing company that until recently was in the public sector
Applications	• Defence-related • Embedded control software • 90% of applications are safety critical	• Very large on-line data processing systems
Development environment	• Variety of advanced approaches, including formal methods & code analysing tools	• State-of-art in using new methods and tools • Keen on using project management methods and tools
Quality framework	• Strong (on the surface), but sub-optimal use of reviews and inspections • AQAP certified (defence quality certificate)	• well used framework • Consistent use of basic quality controls • About to seek software quality certification*
Management framework	• Very complex staff structure • Steep management hierarchy • Low staff morale score • Two year pay freeze • High staff attrition rate	• Simple staff structure • Relatively flat hierarchy • Average staff morale score • Stable and experienced staff group

Table 2.4: Company detail

Table 2.4 gives more details of the two companies, as some QA and measurement success or failure factors may be dependent on company-specific factors, and so the company context is important.

The logistics of this phase of the study follow.

*They still have not achieved this. Unfulfilled 'intentions' to seek certification have been reported in other studies [Neilson & Timmins 1997].

3.4.1 Data re-focusing

The first thing that I did in this part of the study was to re-analyse the data I had already collected on a company by company basis. Table 2.5 gives more details of the study at the two companies. In particular it shows exceptionally high response rates from both companies. It also shows that almost everyone at both companies who was affected by the programmes contributed to this study. There were 24 such practitioners at ES and 125 at IS. (ES implemented measurement in only one critical department, whereas IS implemented measurement across the whole software development function). This enables me to build up a very detailed picture of QA and measurement in both case study companies.

Analysing smaller data sets (such as the ES data set) does, however, have the disadvantage of making statistically significant results more difficult to attain.

Company	Practitioners Targeted	Number responding to questionnaire			Response Rate %
		Total	Manager	Developer	
Embedded Systems	24	20	10	10	83
Information Systems	125	103	48	55	82

Table 2.5: Number of relevant practitioners and response rate

3.4.2 Progress over time

In this part of the study I also tracked the development and progress of both companies' programmes over an 18 month period. I did this by conducting regular formal and informal interviews with QA and measurement managers at both companies. The results of this time-based analysis are discussed in Chapter Eight.

3.5 Phase Five: exploring specific issues

The results of this phase of the study are presented and discussed in Chapter Nine.

In this phase of the study I used the Repertory Grid Technique [Fransella & Bannister 1977] (as described in Section Two of this chapter) to focus further and explore in greater detail practitioners' specific attitudes to QA. I confined myself only to QA because I had difficulty in finding participant companies with measurement programmes to include an analysis of measurement. In this part of the study I particularly follow-up ideas that emerged during earlier phases of the study. These issues included:

1. The variations in attitude of developers and managers.
2. The impact of quality certification.

I predominately collected qualitative data during this phase of the study.

3.5.1 The companies

I recruited a new set of five UK companies for this phase of the study. I believed that 'fresh' practitioners were required, i.e. practitioners who had not been influenced by already taking part in earlier phases of the study. This is a phenomenon usually associated with respondents being influenced by preceding questions on a questionnaire [Berdie & Anderson 1974]. All five companies were approached directly by me after I identified them as potential participants by personal recommendation. While I do not claim this is a statistically representative sample, Table 2.6 does show that the companies in the study range from a very small software house to a very large multi-national company and also cover a wide range of application areas.

Company	*Company Characteristics*			*Software Function Size*	*Main Software Work (ranked)*	*Application Type*
	Type	*Age*	*Size*			
A	Software House	6 yrs	120 staff	120 staff	1. Development 2. Maintenance 3. Support	• Business information systems. • External customers
B	Software House	6 yrs	24 staff	24 staff	1. Development 2. Maintenance	• Technical, real time business systems. • External customers
C	Multinational, electronic goods producer	100 yrs	200 staff in the division	50 staff in the division	1. Development 2. R & D	• Embedded, control software. • External customers
D1[1]	Large, national membership company	100 yrs	3,500 staff	75 staff	1. Maintenance 2. Development 3. Support	• Technical, operations support software. • Internal customers
D2	Large, national membership company	100 yrs	3,500 staff	75 staff	1. Maintenance 2. Support	• Business and telecom systems. • Internal customers

Table 2.6: Company profiles

The companies in the study also had a range of QA maturity levels. Table 2.7 shows that company A had highest quality maturity, having had TickIT accreditation for two years. Company B was least mature, and had no quality programme at all. This meant that for the first time I could explore the reasons why a company had not implemented any formal QA at all.

[1] Companies D1 and D2 are two autonomous divisions of the same company.

Company	Internal Software Quality System		Certified Software Quality System		Plans for Software Quality Certification?
	yes/no	Age	Type	Age	yes/no
A	yes	5 yrs	TickIT	2 yrs	n.a.
B	no	n.a.	none	n.a.	no
C	yes	2 yrs	ISO9001[2]	1 yr	no
D1	yes	1 yr	none[3]	n.a.	no
D2	yes	4 yrs	none[4]	n.a.	no

Table 2.7: The quality context

3.5.2 The methodology

I conducted the study over a three month period during which time I went into each company and:

1. **Collected demographic information on the company:** I did this via a meeting with the quality manager. I collected mainly objective, context setting information during this meeting. The data I collected during this meeting were based around the proforma in Appendix Three.

2. **Discussed quality with a group of developers:** I did this via an informal focus group discussion with peer groups of between four and six software developers. I applied RGT during this meeting.

3. **Discussed quality with a group of managers:** I held a separate, RGT based, focus group with peer groups of between three and four managers. The managers were mainly middle managers, though senior management was represented in the session at the smallest company (company B).

In total I talked to about forty practitioners in ten focus groups during this phase of the study. Participants in these RGT-based discussions were carefully selected by me and quality managers, so as to represent a range of technical specialities and experience levels. The aim being to construct representative groups of practitioners. The samples were not random samples, as I needed to make sure that the people who participated in the sessions would actually talk to me, and this probably introduced a degree of bias into my samples.

During the research sessions I prompted each group of developers and managers to discuss:

---

[2] The processes used to design and produce the final products (consumer durables) are ISO9001 certified, but the development of the embedded software is not specifically certified.

[3] Some functions within company D are ISO9001 certified. Software development is not.

- The existing quality of their company's software.
- How software quality could be improved in their company.
- What QA mechanisms were implemented.
- Whether they thought that their company's quality systems were effective.
- What they thought about quality certification.

This phase in particular allowed me to explore specific issues that had emerged during previous phases. The RGT enabled me to generate some very high quality data on the views and experiences of practitioners to QA. Indeed I found the RGT to be a particularly effective research tool.

The data I collected was fed directly back to all practitioners who participated in this phase of the study.

3.5.3 Evaluating the phase

I had similar problems to those I described earlier when I was collecting data from managers (and as predicated by [Easterby-Smith *et al* 1991]). Manager sessions were much harder to control and keep on track than developer sessions. Managers also seemed more 'self-controlled' in that they tended to be less frank about problems than developers did. They were also much more practised and confident about side tracking and capturing the agenda. However, by this phase, I was much more effective at recognising the phenomena and re-establishing control.

I also had to be very careful with the results from this phase as managers in particular companies were anxious to see the results from developer sessions. They were, of course, less anxious for developers to see the results from their sessions. Managers very much wanted this exercise to be some free management consultancy with their own particular aims and objectives. Resisting this pressure from managers and maintaining their co-operation proved to be rather a balancing act.

4. An analysis of the study methods

Although the methods that I applied during this study are used constantly in social science research, these methods tend not often to be applied in software engineering research. I found that by applying these methods I generated software engineering data of a type not usually collected. This is because data that is collected in software engineering research tends to be product-based data, whereas I collected process and people-based data. I also found that the problems and difficulties that I had in applying these methods seemed much the same as those reported when the methods are applied in other disciplines [Oppenheim 1992]. This section explores some of the issues and problems that I encountered when I applied these study methods.

4.1 Access to data

Gaining access to public data about companies' programmes proved to be a major problem. It made constructing valid state-of-the-art and state-of-practice more difficult than it should have been. Companies are reluctant to publish data in the literature or tell researchers about the real state of the quality of their software or their QA and measurement processes. Indeed I know of many companies which have not made their QA and measurement experiences public. This is probably related to Easterby-Smith's argument that company managers are very aware of the *value* of any data and its possible uses. The reasons for this non-publication have already been discussed.

4.2 Participation and representation

Constructing representative sample populations for this kind of research is of supreme importance. If samples are not representative then results cannot be generalised and conclusions only have limited value to the software industry as a whole. Selective samples have inherent but unmeasurable bias in them [Bryman 1990].

In this study there are two types of representation that I was concerned with:

1. **Representative companies.** Companies participating in the study should representative of the whole software industry.

2. **Representative practitioners.** Practitioners targeted in those companies should be representative of the practitioners in that company.

In most phases of this study I have achieved good practitioner representation within participating companies. This is illustrated by some of the excellent response rates I have already discussed. However, with hindsight, in the companies which I sampled practitioners (rather than conducting a census), I should have been firmer with QA managers about the true randomness of the sample. It is likely that unintentional (or indeed intentional) bias crept into samples in QA managers attempts to provide me with what they believed was a representative sample. In future I would be much more rigorous about adopting true random sampling.

Identifying a sample of companies that is representative of the software industry was more of a problem. This problem stemmed from the difficulty of getting companies to commit time and resources to participating in the study. Consequently I have been less able to carefully select companies. In particular I encountered the following selection problems during the study:

- **Access to practitioners**
  During the research I was also aware of always operating on 'favours' within companies. I was, therefore, sometimes reluctant to push for more access and time with practitioners. This inevitably meant that my research was not as detailed and well followed up as I would have liked.

- **Self-selection**
  The basis on which companies participated in this research means that they effectively self-selected. Obviously self-selection skews the sample in favour of companies who have high quality programmes. Companies who know that they are likely to come out of the research looking bad are unlikely to self-select into such a study. This is one possible reason why I had difficulty persuading companies to participate in the research (i.e. many knew they would perform poorly). However, my results do show that none of the companies that did participate had excellent programmes, so maybe the sample was not too badly skewed. Nevertheless, I suspect that the majority of companies are probably performing even more poorly than the companies which participated in this research.

- **Lack of consistency**
  Ideally, I would have liked participating companies to commit more time and effort to my research. This would have enabled me to collect more detailed data over a longer period of time, and to follow the maturation of QA and measurement more closely. I was able to do this to an extent with ES and IS, but I would have liked to do more of this time-based analysis.

4.3 Scientific validity

In some phases of my study the small samples that I have been forced to work with have made achieving statistically significant results almost impossible. In phases of the study where I have worked with larger sample sizes this has not been a problem. Obviously lack of statistical significance presents results that are not generalisable and, therefore, makes my results less useful to the software industry. However, even where I have achieved good significance levels, the nature of the statistical analysis still mean that it is impossible to prove causal links or measure the strength of relationships. To provide more convincing evidence for such relationships I would really have to conduct a 'before' and 'after' quality and measurement analysis (as recommended by [Oppenheim 1992]).

Furthermore, the sampling and statistical significance problems that I have mentioned mean that I cannot guarantee that repetition of my research on a different (representative) sample of companies in the future would get the same results. Indeed, this test is the final arbiter of a scientifically valid piece of research [Bryman & Cramer 1994].

4.4 Time and resource constraints

Given more time and proper funding this research would have been more rigorous and comprehensive. Research of this nature is surprisingly time and resources consuming. Furthermore many of the most time consuming aspects of this research were basic administration, these included:

- Initially persuading companies to participate in the research.
- Liaising with managers to set up on-site research sessions.
- Travelling to geographically dispersed sites around the country.
- Administering the questionnaire (printing, addressing and posting, data input etc.).
- Preparing and presenting customised feedback to practitioners and companies (as this was often the 'price' of participation).

4.5 Refining the instruments

Many of the problems discussed in association with the pilot study research instruments persisted despite my best efforts to resolve them. I improved questionnaire phraseology in an effort to overcome misunderstandings, only to introduce new misunderstandings. This problem would have been eased had I had the time, resources and participants to pilot all changes to questions as I would have liked. Particular misunderstandings persisted around the meaning of terms such as:

- Organisation
- Department
- Data
- Metrics
- Quality

With hindsight I also wished that I had designed the questionnaires to take more explicit account of causality. In particular been more clear about my dependent and independent variables, and how I was addressing particular hypotheses.

4.6 Future enhancements

Obviously there are aspects of the research that I would improve if I was to re-run this study given more time and resources. These include:

- Iterating the pilot process.
- Conducting more (effective) interviews with practitioners.
- Recruiting more companies.
- Providing more feedback to individual practitioners.
- Repeating data collection periodically to track the development and progress of the programmes.
- Complementing the study with observation-based study methods that look more directly at QA and measurement in practice (rather than simply perceptions and experiences of them).
- Implementing a 'before' and 'after' study, where attitudes before practitioners work within QA and measurement and during and after QA and measurement implementation are analysed.

Chapter Three

The currently reported state of the art

1. Introduction

Reviewing the literature reveals that a fairly comprehensive body of knowledge exists on what is currently regarded as best QA and measurement practice. Furthermore, there is reasonable consensus on what that best practice is. However, much of this has not been validated, and, furthermore, it is skewed towards the tools, techniques and methods to implement, rather than the implementation process itself. There is an obvious gap in the literature as far as well validated approaches to implementing QA and measurement goes - this thesis addresses that gap. However, there remains some very sensible and worthwhile advice given by the experts in the literature.

I discuss the reported state-of-the-art in this chapter. In the next chapter I analyse the experience reports which have been published by companies in the literature, and establish a current state-of-the-practice for the implementation of programmes in the software industry. In the following chapters I discuss the state-of-the-practice that I observed in the companies in this study. In this thesis I compare what I found in practice, to what is reported as being practised, and to what is regarded as state-of-the-art in the implementation of programmes.

To focus effectively on the state-of-the-art in successfully implementing programmes I discuss implementation theory as described in the social, organisational and management science literature. This body of knowledge is important, as the theoretical underpinnings of any effective software QA and measurement programme must be drawn from the principles already established within generically applicable implementation theory. If the software industry does not use this as its base then the implementation of QA and measurement will be very weak and the programmes much more liable to failure.

During this study I found that the software industry seemed reluctant to use knowledge from other disciplines, and consequently many areas of the QA and measurement literature failed to incorporate existing and validated implementation theory. Obviously in reinventing the implementation wheel, the software industry is creating an additional and unnecessary obstacle to progress in this area.

The chapter is structured as follows: In Section Two, for completeness, I discuss the state-of-the-art in QA and measurement methods and techniques. In Section Three I continue in this theme by describing the state-of-the-art in software measurement methods. In Section Four I present the state-of-the-art in generic organisational theory. This provides the basis on which the implementation of tools and methods should be based. In Section Five I discuss implementation best practice as described in the software QA and measurement literature. In this section I describe every aspect of an optimised implementation process as reported by the experts. I conclude in Section Six.

Where I report on work published from within specific companies, I refer to those companies only by name. In Appendix One I provide a key which gives the full reference for the work cited.

2. State-of-the-art QA methods

Although, in this study, I am predominately interested in *implementing* effective QA methods, in this section for completeness, I provide a brief overview of the QA methods currently considered to be state-of-the-art. I also attempt to set the boundaries to this study by defining various quality related terms. I describe a range of different QA and methods that companies implement. However, time prevents me from including an analysis of:

a) Specific aspects of software development that are critical to quality. For example: testing, configuration management etc. are beyond the scope of this thesis.

b) The extensive range of valuable and increasingly popular tool support for QA activities (for example, testing tools, static analysers etc.).

Although more approaches to QA exist than I have included here, I feel that there are some spurious methods that only serve to confuse developers (and make them cynical of 'quality'). Indeed QA suffers from a reputation of 'hype' and so I have tried not to perpetuate this by describing some of the methods that I consider to be wishy-washy and detracting (for example, the TQM movement [Pike & Barnes 1996]).

2.1 Defining quality

A major problem in the area of software quality is a lack of consensus on the definition of quality. This basic ambiguity makes studying other aspects of quality more difficult than it might be.

A wide range of definitions of quality exist, many of which indicate fairly intransigent 'camps' of opinion. Two definitions that are common, are:

1. *Fitness for purpose*: producing software that really solves the problem it was meant to solve and in a way that is acceptable to the user.

2. *Conformance to requirements:* producing software that is to specification with no defects.

Other definitions that are highly attractive to industry are based around timeliness and delivering systems on time and in budget.

For a comprehensive overview of defining software quality see the editorial of IEEE Software's special issue on software quality (IEEE Software, January 1996).

2.2 Defining QA

It is important to distinguish between quality and quality assurance, as the terms are often, wrongly, used interchangeably. Indeed people often mistakenly confuse QA for quality. It is my view that it is possible to have quality without QA and QA without quality.

A variety of QA definitions exist, the most significant of which include:

> *"the activity of providing all concerned the evidence needed to establish confidence that the quality function is being performed adequately"* [Juran *et al* 1979]

> *"a planned and systematic pattern of all actions necessary to provide adequate confidence that the item or product conforms to established technical requirement."* [IEEE 1983]

For the purposes of this study I take a wide ranging view of QA and consider it is all the actions necessary to produce a high quality product. My view is an extension of the IEEE definition, as I am interested in more that delivering a system that conforms to technical requirements. Such a view of QA for software inherently means that QA is inextricably linked to the software development process and means that the quality of the development methods, techniques and tools applied during that process are critical to quality. However in this thesis I restrict myself to those aspects of QA that are QA-specific and are not part of the traditional software lifecycle.

2.3 Widely used QA techniques

In this section we review briefly the most well cited QA techniques.

2.3.1 Reviews and inspections

Reviews and inspections are cited throughout the literature as the cornerstone of QA programmes. Indeed they are probably the development technique with the most quantitative evidence to recommend them. Formal inspections (Fagan inspections [Fagan 1976]) are particularly widely regarded and recommended (see, for example, [Gilb & Graham 1994]). Furthermore, not only are inspections and reviews of direct importance to the quality of the software product, but they are also a very effective tool for process improvement.

However, despite the convincing evidence in favour of reviews, in practice they are reported as being used less frequently than operational standards are used (coding standards, documentation standards etc.). The cost of implementation seems to be the most significant reason for companies not implementing them (despite the benefits they are shown to have).

2.3.2 QA standards

Standards are widely regarded as the basis of any effective QA programme and are largely process-based (though standards defining products do exist, for example, the software quality standard ISO 9126 [ISO 1991b]). Those who argue in favour of standards maintain that a high quality process (usually defined by standards), will lead to a high quality product. Although this argument seems intuitively correct there is no convincing scientific evidence to support it. Indeed there are those who argue that process-based standards are weak as they do not explicitly assess the quality of the final product [Fenton *et al* 1994].

The implementation of low level operational standards and procedures is regularly recommended in the QA literature (for example, [Gillies 1992], [Hambling 1996]). Such standards include design, coding and testing standards, but may also be procurer-led (for example, the MOD and DOD standards). Furthermore, because well designed standards and procedures capture best practice and ensure that this best practice is always practised, they are cited as the basis of process improvement and the way in which a repeatable process is sustained [Dion 1992]. Indeed some of the state-of-the-art industry QA programmes cite standards, procedures, reviews, audits and measures as the core of an optimised improvement programme (for example, the NASA/IBM programme [Billings *et al* 1994]).

Despite there being no shortage of purported best practice standards (an estimated 250 in software development), [Fenton 1996] believes that there remains no consensus on what best practice actually is. Indeed in his work he has standards with similar objectives that recommend very different practices and many standards that are not standards at all but codes of practice or guidelines that are impossible to either enforce or measure compliance to. He concludes from this that software development is not yet mature enough for standardisation.

Furthermore, the SMARTIE project [Pfleeger *et al* 1994] and other research has been very critical of the real impact of standards. Such criticisms include that compliance to standards is not checked for and results in standards not actually being used. However, more profoundly, the impact on software quality of using standards is yet to be proved.

Despite some criticism regarding their maturity, standards and procedures remain a crucial element of most QA programmes. Indeed the popularity of a number of certified standards seems to be growing. I discuss some of these below.

2.3.3 The ISO 9001 family of standards

ISO 9001 is a generic quality standard which can be applied to the quality management systems of companies in any industry sector. From this standard ISO 9000-3 [ISO 1991a] guidelines have been developed which applies ISO 9001 specifically to the development of software. Furthermore, TickIT [Dti 1992] provides published guidance on the application of ISO 9000-3. All of these standards are process orientated and all are currently more popular in Europe than in the US. However, it is possible that when SPICE [Barker 1995] is completed these standards will be usurped in Europe.

ISO 9000-3 is currently probably the most sought after software kitemark in Europe. Most companies believe that there is either an indirect or direct marketing advantage to having the certificate. However, there is also a great deal of criticism about the efficacy of the certificate. Indeed probably the biggest criticism levelled is that, although process quality is addressed, product quality is overlooked entirely. Consequently it is widely believed that the standard allows the consistent production of poor quality software.

2.3.4 The Capability Maturity Model (CMM)

The CMM [Humphrey 1989] is increasingly emerging as the most significant process assessment model in the world. Indeed it has had a major impact on software QA procedures. In assessing the quality of companies' software development processes it not only awards a maturity level to that process, but it also gives companies detailed pointers to the kind of QA mechanisms that a mature process should have in place. Consequently the model has rapidly grown as the preferred process assessment certification in the US and this seems to be rapidly spreading to the rest of the world. However, again it does not explicitly consider the quality of the product, concentrating entirely on the process, furthermore some people argue that the CMM is highly political and US culture-specific.

3. State-of-the-art in software measurement methods

I now discuss the most significant measurement techniques and methods reported in the literature.

3.1 A scientific basis to software measurement

For many years a wide variety of individual software measures have been designed by researchers and, occasionally, implemented by companies. However, many of these measures have lacked any theoretical measurement rigor. The data collected via such measures was, therefore, highly unreliable and its use counterproductive. Norman Fenton was the first to systematically recognise and address this issue. His influential 1991 book 'Software Metrics: A Rigorous Approach' [Fenton 1991] and its follow-up [Fenton & Pfleeger 1996] set about creating a comprehensive and disciplined framework for measuring software products and processes. Most significantly he successfully injected the mathematical discipline of measurement theory into the measurement of software - this had largely been overlooked beforehand. This has had a significant effect on the software measurement community. Furthermore, it shows that it is possible for software engineering to successfully implement generically valid principles from other disciplines. I hope, therefore, that in the way that measurement theory has been accepted within the development of software measures, implementation theory can be accepted within the implementation of those measures.

3.2 Goal Question Metric (GQM)

GQM [Basili & Rombach 1988] is an established approach to specifying goal-orientated measures. It particularly addresses the very real problem of companies not knowing what to measure and as a result, collecting data that they have no real use for [Hetzel 1993, McGarry 1996]. GQM is very popular with companies and commentators [Fenton 1991, NCC 1993, Kitchenham 1993, Rombach 1990, Möller 1993, McGarry 1996, Rubin 1995, Daskalantonakis 1992]. Indeed the use of GQM has been shown to be effective in many companies, most notably NASA/SEL.

Although GQM is generally a well respected approach to measurement and one that is regularly recommended in the literature there remains criticism of it. This can be categorised according to the following.

3.2.1 Goal setting

This is reported as very difficult. There is uncertainty about who in the company should set goals and define measures and different companies have very different views on who should do this. Most set goals top-down, *i.e.* managers define the goals [Carlson & McNurlin 1992]. However, a few other companies believe that practitioners are in the best position to set realistic goals, and so their goals are formulated in a bottom-up fashion. Indeed, [Bache & Neil 1995] were so critical of GQM that they propose the MQG (Metric Question Goal) approach. This consists of:

1. Measurement: Collect and analyse data.
2. Questions: Why results are as they are and what improvements are needed.
3. Goals: Set according to known problems, rather than on manager's subjective view of what is wrong.

This MQG model is similar to Bill Hetzel's notion of bottom-up measurement [Hetzel 1993]. Hetzel believes that companies should select from well established and easy to collect measures rather than informally defining their own measures via GQM and thus identifying things that might prove difficult to collect. In this way the measures selected will have a better theoretical basis. Hetzel also believes that GQM's top-down nature results in weakened support for measurement at a practitioner level. Hetzel advocates a bottom-up measurement process which starts with measurable engineering points from which goals are built. Thus the engineering process is supported, rather than the management process and means that data becomes a natural by-product of the engineering process and will, therefore, have practitioner involvement and support. Indeed, Hetzel reveals a good appreciation of the human issues that can make or break a measurement programme.

3.2.2 Inadequate understanding

Some commentators believe that implementing GQM is problematic because of the generally poor understanding of the specific software development problems demonstrated by the industry. Software problems are not well understood, well defined nor de-composed and this makes identifying sensible goals and measures difficult. GQM also assumes that companies identify the correct goals and the correct measures to measure those goals, and in fact this is very difficult to do in reality.

However, most of these problems are issues of implementing GQM rather than the technical correctness of the model. Consequently if a rigorous approach to implementation is taken these problems should be less significant.

3.3 The Hewlett Packard approach

Perhaps the most important influence on the implementation of measurement has been the publication of [Grady & Caswell 1987]. The book (and its [Grady 1992] follow-up), describes the Hewlett Packard software measurement programme. In doing this the 'HP Way' of implementing measurement is described. Indeed, Grady and Caswell identify many of the important (but usually neglected) organisational issues surrounding software measurement.

The 'HP Way' is based on a measurement council, of about twenty respected (nominated) managers and engineers, who:

1. distribute information;
2. propose improvements;
3. attend meetings and share results;
4. actively improve the process.

More significantly the 'HP Way' proposes a ten point method by which measurement should be implemented:

1. Define objectives for the programme;
2. Assign responsibility;
3. Do research;
4. Define initial measures to collect;
5. Sell the initial collection of those measures;
6. Get tools;
7. Train;
8. Publicise success and encourage the interchange of ideas;
9. Create a database;
10. Establish a mechanism for changing things.

Clearly the 'HP Way' of implementing new mechanisms is as relevant to implementing QA techniques as it is to implementing measurement.

However, although companies have used this approach to implementing measurement there are problems with it. For example, Racal did not find its adherence to the "HP Way" totally satisfactory. They discovered that the approach did not "adequately reflect Racal's organisational outlook" and so were forced to customise the Hewlett Packard approach. Indeed even Hewlett Packard had initial difficulties with their training programme, as the training presented was too theoretical and not practical enough.

3.4 ami

The ami method [Pulford *et al* 1996] is a step-by-step approach by which companies can implement an effective measurement process. It is a clear and practical way in which measurement can be implemented by ordinary practitioners in ordinary software companies.

The approach to measurement was developed by an industrial consortium led by South Bank University and has proved very popular amongst practitioners (for example, GEC Marconi, Pearl Life Assurance). It combines CMM and GQM, first the SEI CMM questionnaire is used to establish process maturity, then GQM is used to identify metrics which are appropriate for the maturity level. As a result a measurement process is developed that is company specific.

ami does not, however, really address the complex organisational issues surrounding the actual implementation of the measurement process that it helps to design.

3.5 Some common software measures

A wide range of ways in which software can be measured exist. Many software attributes can be measured either directly or indirectly, and these can be categorised as either *product, process* or *resource* measures [Fenton 1991]. Indeed, an effective measurement programme will include a balanced range of these measures.

An extensive analysis of specific measures is outside the scope of this thesis, as I am interested in how companies go about implementing measures rather than what measures they implement. However, for completeness, I present the following overview* .

3.5.1 Complexity measures

It is widely believed that complex software is poor quality software. Complex software is typically more error-prone and more difficult to test and maintain than non-complex software. Consequently, companies are very keen to reduce the complexity of their software, and so the measurement of complexity is

* I have broadly adopted Norman Fenton's taxonomy of software measures, a full analysis of which is available in [Fenton & Pfleeger 1996], an overview of which is available via http://www.csr.city.ac.uk/~nf.

popular. However, complexity can only be measured indirectly because it is an aspect of software that is made up of many individual attributes. This means that complexity is difficult to measure accurately and objectively. There have been many complexity measures proposed over the last twenty years. Many of these measures (especially the older ones) attempt to measure only the complexity of code. The most famous measures of complexity include:

- **Halstead's Software Science metrics**

  Analysing the occurrence of operands and operators in code is the basis of this measure.

- **McCabe's Cyclomatic number**

  The flow of control through a program as dictated by conditional statements is how complexity is measured here.

Both of these measures have been heavily criticised for taking too narrow a view of complexity and for having a naive view of it. Furthermore, both are said to be empirically and theoretically flawed [Fenton 1991]. However, despite this criticism both measures remain popular.

3.5.2 Resource estimation models

One of the most significant problems that companies have during the development of software is in accurately predicting the time and effort necessary to complete a project. As a result a great deal of effort has gone into designing measures and models to aid the estimating process. The best known of these is COCOMO [Boehm 1981]. By analysing a project in terms of a number of key indicators, COCOMO produces an estimate of the amount of effort that project requires. Although the model is quite sophisticated, again, it has had a great deal of criticism levelled at it. The most significant criticism is that it depends on an initial subjective estimate of the size of the project.

3.5.3 Functionality measures

Functionality measures attempt to measure software size in a way that is more valuable than the conventional counting lines of code. Albrecht's Function Points [Albrecht & Gaffney 1983] is the most well known of these measures and is used increasingly by industry. Function points can be counted from the specification and are determined from an analysis of various aspects of the system, including how many inputs, outputs and file accesses that the system must process. The technical difficulty of the system is then assessed by rating the system against fourteen technical factors. Again, despite the popularity of function points (especially for data processing applications) they have been heavily criticised for the amount of subjectivity their calculation depends on.

4. The state-of-the-art in organisational theory

There is a huge body of well established and validated knowledge within the organisational and management literature. Much of this material is of direct relevance to the development of rigorous QA and measurement implementation strategies. In this section I summarise those elements of organisational and management theory of most direct relevance to this thesis.

In this section I identify those aspects of implementation theory which provide the backbone to a company's implementation strategy. It is however, important to understand that implementation theory is not a *silver bullet.* It is also not an *off-the-shelf* list of *do's* and *don'ts,* it is more substantial than that. It provides an underlying rationale which should be used to construct a cogent implementation strategy. It does not, however, provide that strategy. It is anomalous to what measurement theory does for software metrics. It provides a basis of understanding that should be used to construct individual implementation strategies for particular changes in particular companies.

Furthermore, implementation theory is broad. It includes aspects of the theory of culture, norms, psychology, behaviour and motivation. Although none of these areas alone are directly applicable to the implementation of QA and measurement, a basic understanding of them enables company's to construct rigorous low level implementation strategies. Without this background implementation strategies are more likely to be unstructured and lacking a clear rationale.

4.1 Organisational culture

4.1.1 Types of culture

Different types of organisational cultures exist, in which different implementation and change strategies will be most effective. Although companies can be classified according to a variety of taxonomies, the following taxonomy [Wilson & Dawson 1996] is of most relevance to this thesis:

- **Closed**
  These organisation have a strong top-down management style. They are very hierarchical and insist on clear, distinctive and rigid organisational roles and responsibilities. Successful change in these types of company is dependent on a very formal and planned approach that is well documented. However, these kind of companies are the most difficult in which to implement effective change [Kanter 1984]. They tend to be highly segmented companies in which segments operate independently

of each other and their behaviour becomes narrow and rigid. In such companies there is little incentive to change (or share experiences and good practice or even to communicate) as this might upset the configuration of the organisation. Such companies also tend to be very *political,* and this can damage change [Wilkinson & Witcher 1993]. Despite this type of organisation being common, they are essentially disempowering and non-innovative companies.

- **Random**

  These companies are based on informal groups working under light and flexible management. Individuals in such companies are likely to have undisciplined ways of working as there will be few systematic methods in place. Successful change in this type of organisation is reliant on personal influence and persuasion rather than top down directives. *Individuals* are empowered in random companies.

- **Open**

  A collaborative and sharing culture exists in *open* companies. This culture is also supported by a flexible and effective organisational framework. *Groups* and *individuals* are empowered in these companies and so effective change initiatives must be highly participative.

In this research I encountered companies of all three types.

4.1.2 Cultural change

Although effective change interventions must be planned according to what is going to be most appropriate given the existing organisational culture (as new practices that are compatible with existing ways of doing things are most likely to be successful [Constantine & Lockwood 1994]), it is also possible to try and change the underlying organisational culture*. The prevailing culture in an organisation is embodied by the shared values and beliefs within that organisation. These are based on the culture messages sent from management. Management policies, plans and behaviour all establish and reinforce an organisation's culture [Rouse & Watson 1994]. This suggests that to change the culture of an organisation, management must change the way in which they manage. Again, this reinforces my thesis that it is within the grasp of managers to manipulate an environment so that the implementation of QA and measurement is more likely to be successful. Success or failure is not a random process, it is a controllable and manageable process.

* There are, however, two schools of thought on how difficult engineering an organisational culture change is: the *integrationalist view* [Myerson & Martin 1987] and the *differentiation view* [Martin & Myerson 1988].

4.1.3 Sub-cultures

Another aspect of organisational culture relevant to my thesis is that *subcultures* exist in many companies, and that these often result in the existence of *official* and *actual* working practices [Neilson 1994]. This is a phenomenon often anecdotally mentioned (and occasionally mentioned as an issue in the software engineering literature for example, [McGarry 1996]), and one which I saw direct evidence of in this research.

4.1.4 Cultural behaviour

The [Neil 1994] analysis of bureaucratic dysfunction is also relevant to this thesis as it again explains some of the behaviours that I found in this research (these ideas have their foundation in Weber's classical organisational theories [Weber 1947]), and include:

- Rules originally intended to serve an organisation becoming important in their own right.
- Roles and status become de-personalised and lead to rigid behaviour.
- Rigidity leads to damaged customer relations.
- Standardisation and routine make change and adaptation difficult when circumstances change.

Neil also cites [Gouldner 1954], who also analysed patterns of bureaucratic behaviour, which again proved relevant to my research findings. He identified three basic patterns of bureaucratic behaviour:

- **Mock**

  Rules and procedures are imposed from outside. Lip service is then paid to these rules and they are ignored. People then practice their own rules and procedures.

- **Representative**

  Rules are well designed and are implemented and adhered to by managers and employees because those rules are accepted as valuable.

- **Punishment-based**

  Rules are imposed where deviance from them involves the application of sanctions.

4.2 Motivating practitioners

4.2.1 Changing behaviour

QA and measurement requires practitioners to work differently. Whether individuals are willing to change in this way depends on two things [Anderson & Kypnanou 1994]:

1. How satisfied they are with their current situation.
2. Their perceptions of the risks to them personally of the change.

However, *'No one can be forced to care about something'* [Dell 1989], and so practitioners must be *motivated* to co-operate in programmes. There are many aspects of motivation theory that are relevant to this aim. Indeed the overall aims of motivation theory are very relevant to the successful implementation of QA and measurement, [Ford 1992] describes the aims of motivation theory as being to rationalise:

1. Where people are personally heading and what they are trying to achieve (the direction of behaviour).
2. How people get turned 'on' or 'off' things (the energisation of behaviour).
3. Why people decide to try something, stick with it or give up on it (the maintenance and sustenance of behaviour).

A very important aspect of motivation theory that is very relevant to my thesis is the notion that people behave rationally in the systems they are operating within. Individuals attune their behaviour to co-ordinate with the predominate behaviour they observe around them. In a work situation this means that individuals behave in a way that conforms to the way that others in the organisation behave i.e. they conform to the culture within an organisation. This is a very important principle to my thesis, as it adds immediate weight to my proposition that if QA and measurement is managed effectively, and operates within a positive organisational culture, then the behaviour of practitioners within that company can be managed, and the success of the programme can be controlled.

4.2.2 Goal driven behaviour

Another important aspect of motivation theory is the suggestion that individuals practice '*context-specific, goal-directed patterns of behaviour*' [Ford 1992] and that this behaviour continues until either:

- A goal is accomplished.
- Another goal takes precedence.

- A goal is evaluated as unobtainable.

In the context of an increasingly goal-orientated approach to software quality, these patterns of behaviour are significant. Also of relevance to companies setting software quality goals, is that motivation theory says that motivating humans effectively requires:

> '*...strategic emphasis on attainable short-term goals combined with a periodic review of the long term goals that give meaning to one's short term pursuits.*' [Ford 1992, p99]

Motivation theory also addresses the situation where organisational goals are not aligned with the goals of individuals. Strategies suggested to overcome this goal incongruence include:

1. Effectively communicating organisational goals to individuals. Many individuals are said to be happy committing to organisational goals, and so a failure to communicate those goals is unnecessarily damaging - *"Invisible goals are usually hard to hit"* (Tom Gilb).
2. Applying a variety of goals.
3. Selecting workers with goals compatible with the companies' goals.
4. Managers must be:
    - enthusiastically committed to organisational goals;
    - personally focused on the goals;
    - reinforcing those goals by defining and measuring progress towards them;
    - communicating goals to workers at every opportunity.

Management fairness and trustworthiness is said to be the foundation of effective motivation.

In this research I found that, although improvement goals were very important to companies, they were failing to implement goals effectively (see Section 5.1 in Chapter Six).

4.2.3 Motivating for desired behaviour

An effective motivating intervention must be:

1. aware of peoples' personal goals;
2. within the capabilities of the person;
3. realistic, given the resources that are available.

Approaches to motivating desired behaviours are numerous, the most relevant to my thesis include [Dell 1989]:

- **Facilitating goals**

  Goals must be clear and understandable. Interventions must align multiple goals so that they are not in competition with each other, prioritisation of those goals may be necessary. Explanations must be given that show how goals might be achieved.

- **Feedback**

  Progress towards a goal cannot proceed effectively without feedback, as people do not know how well they are doing.

- **Flexible standards**

  Goals must be attainable, if they are not then this is very de-motivating. Consequently, standards must be adjusted if they prove too challenging. Goals must be set according to the prevailing circumstances. Standards should be increased in small, manageable increments, and be at a level that is challenging but attainable.

- **Direct evidence**

  People may need direct evidence that other people can achieve a certain goal. This may take the form of direct observation of other people, or learning/training opportunities.

- **Do it**

  The first step of a motivation intervention must involve a small, safe step, not a transformational change. It is also acknowledged that there are many different means to one end and so if one intervention does not work then a different one might.

- **Collaboration**

  People need to be involved in goal setting, so that current and core concerns are addressed. Impersonal goals set by managers, with which workers are expected to align themselves will not be successful in the long term (despite them sometimes appearing successful in the short term).

4.2.4 De-motivating for desired behaviour

Approaches that are likely to de-motivate desired behaviour include [Dell 1989]:

- managers asserting authority and initiating a power struggle;
- interventions that are counter to peoples' personal goals;
- coercion and manipulation: threats, punishments, rewards and forced competition. Although these may produce short term benefits, they are likely to be detrimental in the long term. Rewards, in particular, are (arguably) counterproductive as they serve to undermine the genuine importance of the original goal.

The work done by [Keinayel *et al* 1991] on motivating programmers suggests that managers in the software industry are particularly bad motivators. My research supports that finding.

4.2.5 Changing behaviour

Implementing QA and measurement changes a working environment. Formal theories of change are, therefore, very important to implementing successful change.

[Handy 1993] explains the responses people have to change as being one of three possibilities:

- **Compliance**
  This can happen when individuals comply because something has been imposed from higher in the organisation (hence the importance of clear communication and information for people likely to react in a compliant way). However, this will only occur in a highly structured organisation that has lots of rules in place and where compliance to the rules is regularly checked. Although quick and effective in the short term, this approach will result in ill-feeling and reluctance in the longer term. It will then generate extra work to ensure continued compliance.

- **Identification**
  A person can be influenced so much by another person that they adopt the views of that person. This is most likely to occur in random companies. However it makes change dependent on one person. Experts/champions/gurus can fulfil this role.

- **Internalisation**

  Here people accept ideas as their own together with the associated change. This is most likely to occur in an *open* organisation and is the most effective form of change.

In this research I found no company had any clearly planned change strategy.

4.3 Management behaviour

Successful change is dependent on how change is managed. Consequently the behaviour of managers during the implementation and maintenance of QA and measurement is crucial.

4.3.1 Middle management

One of the most pernicious factors in any change is the impact of middle managers. [Kanter 1984], found that middle managers block innovation and change. She believes that this is because, although middle managers only have a relatively small amount of organisational power, they want to cling to that power and they perceive that change threatens that power. Senior management, on the other hand, have much more power and are, therefore, less protective of it. Consequently, change initiatives can be undermined by the behaviour of middle managers. Senior management may espouse good ways of implementing change, but if middle management covertly ignore these approaches, the change initiative is likely to be unsuccessful. Any strategy for change must, therefore address the potential reluctance of some middle managers.

Problems with middle managers are regularly highlighted in the software literature and several commentators (for example, [Fenick 1990]) suggest that some short term benefits must be built in for middle managers. This problem is effectively summed up by [Davidson 1991]:

> *"People in the middle tend to be most threatened, people at the top can see the benefits in terms of profits. People at the bottom can see the benefits in terms of getting a better idea of how big the programming task is, how clever their code is and so on. For the people in the middle, measurement can be a stick to be beaten with"*

Indeed I uncovered evidence of just this happening (see Chapter Nine).

4.3.2 Management strategies

- **Top-down implementation**

  The management in some companies implement change in a strictly top down way using management dictat. Again [Kanter 1984] found that this approach is unlikely to be successful in the long term. I found evidence to suggest that such an approach is common in the software industry. Many reports on programmes describe a very top down approach and this may explain the relatively poor performance of these programmes.

- **Leadership**

  In any change initiative management at all levels must be *seen* to take a leading role and *demonstrate* commitment to the change, not just paying lip service (something that is reported frequently in the QA and measurement accounts and something I found evidence of). [Porter & Parker 1993] found that it was particularly important that senior management were seen to be *driving* the process of change.

- **Communication and participation**

  Organisation theory researchers [for example, Porter & Parker 1993] have found that *communicating* to practitioners about change plays an important role in the success of change efforts. Internally marketing changes by publicising achievements and recognising contributions were particularly effective. Furthermore, communication combined with practitioner participation was found to be a key determinant of success. Participation in the definition and implementation of changes, played a significant role in minimising resistance to change and, therefore, maximising success. Porter and Parker observed companies using a range of employee participation levels in change efforts, from optional to mandatory, and found that both extremes were sub-optimal. Indeed, the most effective change efforts that they observed *motivated* people to *want* to become involved.

4.4 The impact of measurement

My belief that the principles behind successfully implementing any change are not industry nor innovation-specific is fundamental to this thesis. I, therefore, argue that generic implementation strategies are relevant to implementing QA and measurement programmes in software development environments. However, measurement is slightly special, as some of the reactions that people have to some aspects of measurement in the workplace are unique. Measures that either are, or can be perceived to be, performance or productivity related are particularly problematic. Although managers have been urged by many commentators to avoid collecting data that can be attributed to individuals (for example, [Grady & Caswell 1987], [Pfleeger 1993]), such measures are still often collected. Furthermore, even when data on individuals is not collected, measurement can still be perceived to be very threatening. Uncertainty about how data will be used makes people fearful and resistant.

Fear and resistance is probably a reasonable reaction as, even in Hewlett Packard [Grady & Caswell 1987] managers were reported as *'having the urge to use data against people'* and even some high profile measurement commentators urge dubious measurement practices. [Card 1988] suggests that one benefit of software measurement is that people "perform better when they know they are being measured". Furthermore, [Ormrod 1990] suggests that measurement data is useful for performance related pay schemes as well as comparing the performance of different teams. Again he misses the point that the process and product should be the prime focus of measurement, if that measurement effort is not to be damaged and damaging in the longer term.

However most worrying is a statement from [White 1993], who said:

> *"Only those who do not know how to change fear measurement".*

To try and counter some of the potential damage of measurement the discipline of productivity measurement has experience and knowledge that the software industry would benefit from.

Of particular importance is the evidence [Tuttle & Sink 1984/5] suggesting that measures that *can* be used to assess performance, will result in people feeling the need to make the figures look good, which will result in corrupt and misleading data being collected. This will be the case whether or not the data is *intended* to be used for assessment purposes, and will also be the case whether the data is based on individuals or workgroups. Indeed in this research I found evidence suggesting that one of the few pieces of published QA and measurement advice that many software companies heeded, was to avoid collecting data attributable to individual practitioners. However, I found that this led to *managers* making the data which related to their *work group* look better, rather than individual practitioners manipulating data about themselves.

[Tuttle & Sink 1984/5] suggest the following aspects of measurement that individuals are fearful of and, by implication, an effective QA and measurement strategy should account for:

- **Misunderstanding or misuse of measures**

  People worry that because managers may not know enough about day-to-day activities, they will not have enough contextual information to use the data appropriately and fairly. This factor is well understood by state-of-the-art software development companies, and is a principle often discussed in the measurement literature (for example, [Hetzel 1993]). It is also probably the motivation behind commentators urging companies to gain a deep understanding of the software development process before introducing measurement [Pfleeger 1993].

- **Exposing inadequate performance**
  This is especially problematic where goals are undefined or ill-defined, and people do not know what they should be aiming for. In these circumstances people are particularly reluctant to expose their work. Again, software measurement commentators acknowledge this issue [Hetzel 1993].

- **Additional time and reporting demands**
  Data collection that is perceived to, or actually increases paperwork and therefore takes more time will be unpopular.

- **Distortion of performance**
  Data collection is not just a passive mirror of the situation. The measures selected will send signals about what is being valued. Furthermore, if unimportant factors are measured then these are the things that people will put effort into. Measures should be a function of goals.

- **Reduction of autonomy**
  Highly skilled technical people usually want autonomy. It can be argued that measurement reduces that autonomy, as it concentrates effort towards defined goals. If there is disagreement about the validity of those goals then problems can result.

Although I found that many principles of best implementation practice were not used (even in some state-of-art QA and measurement programmes) measurement was, on the whole, handled better than the implementation of other QA techniques. Furthermore, many aspects of good measurement practice are reflected in the recent major software measurement initiatives (for example, [Pulford 1996], [Pfleeger 1993], [Basilli & Rombach 1988]).

5. The state-of-the-art in QA and measurement implementation

In this section I report the results of my analysis of the software QA and measurement literature. In this analysis I discuss what the experts say about specifically implementing QA and measurement. In this section I present the state-of-the-art in implementing software QA and measurement.

In this section I cite the experiences of many software companies. These company experiences are described in detail in the next chapter, and full reference to these companies is in Appendix One.

5.1 Company culture

A company culture that is focused on improvement is the key determinant of successful programmes. For example, in Motorola's very successful programme [Daskalantonakis 1992], cultural and human issues were found to be fundamentally important, and ones which need to be addressed 'up-front'. Developing an 'improvement mentality' was critical to the success of that programme.

Despite most company reports and commentators in the field agreeing on this, there is little proper discussion about how companies can actually achieve an improvement culture. Specific examples of what companies must do are thin on the ground. Again, it seems to be a case of paying lip service to the importance of company culture, but then putting most effort into *what* to implement, and overlooking the critical success factor of *how* to implement it. This situation, again, illustrates the industry's lack of awareness and practice of organisational and management theory, where cultural issues are regarded as paramount.

[Neilson & Timmins 1997] have analysed organisational commitment to improvement in Australia's software industry. Not only do they confirm that a company's commitment to improvement is the cornerstone of success, but that commitment to a real, tangible improvement culture was rare in companies. Indeed they found that often the companies that say most about quality do not do very much about it. An example of this is one of the companies they studied talking for eighteen months of its intention to gain quality accreditation, but doing nothing towards achieving it. One of the companies that I studied also did exactly this.

5.1.1 Management commitment

Securing real management commitment is a major step towards a company developing an improvement culture. It is often the case that managers just pay lip service to improvement and actually change nothing. This is especially the case of middle managers and something that I found evidence of in this research.

It is, however, crucial to gain top level support for QA and measurement. If this is missing then it will not be taken seriously by the rest of the company (Siemens), and, moreover, it will be seen as an easy target for financial cuts (Racal).

One of the ways that reports suggest for companies to achieve and demonstrate senior management commitment is by assigning a senior high profile advocate to QA and measurement. This responsibility for overseeing the programme must be assigned to someone senior in the company, as the status of the person selected will reveal to the whole company the seriousness of the implementation (Hewlett Packard, Siemens).

5.1.2 Practitioner involvement

Practitioner involvement is also reported as crucial to the successful implementation of QA and measurement [Hetzel 1993, NCC 1993, Möller 1993, Humphrey 1989]. This appears heartening, as it is a key implementation success factor emphasised in the organisational and management literature. However, closer analysis of the accounts that report this as important, tend to be those written by commentators, rather than those directly reporting on specific company programmes.

It seems that companies have still not appreciated the importance of practitioner involvement. This may be related to my finding that very few companies actually do any proper research before implementing programmes and so do not know how important this factor is. Many companies make the mistake of assuming managers are best placed to design programmes. However, the literature makes it clear that practitioners are in the best position to know what is practical and realistic to implement, and what will be non-threatening and, therefore, acceptable to practitioners.

My suspicion that practitioners are not involved enough is confirmed by several reports which warn against implementing purely management driven QA and measurement. [Neilson & Timmins 1997] found that many quality initiatives were handed down 'on high' from managers and that these were viewed with suspicion by developers. Developers believed such initiatives were politically motivated and were not about improvement at all. In response to this Neilson and Timmins also found that it was possible for developers to appear committed to QA and measurement, but not actually be (they too could just pay lip service). Furthermore they found that managers assumed they would easily gain staff commitment and were surprised by negative reactions. Managers in their study expected developers to change very quickly and they let the strength of their personal commitment blind them to reality. [Hetzel 1993] also implicitly highlights this manager blindness, when he warns practitioners against the temptation to just tell managers what they know they want to hear about quality.

One of the reasons that companies informally give for failing to gain practitioner commitment is that practitioners will not like QA and measurement. However, I and others have found that contrary to this, it is common for lots of informal QA and measurement efforts to be already taking place, none of which have any operational definitions or standards (Boeing, EDS General Motors). It seems that if there is no proper company-wide QA and measurement programme, individuals and groups will start up their own informal efforts - people like working within a structure [DeMarco 1982].

The failure of many programmes is attributed by many commentators to companies not involving practitioners sufficiently in the development of programmes. Indeed Robert Glass believes that many software crisis-type problems are the result of no one listening to practitioners [Glass 1994]. [Dion 1992] also believes that too many companies spend too much time documenting methods and not enough time ensuring that those methods are 'embraced' by practitioners.

A notable exception to this lack of practitioner involvement is Raytheon, where 'practitioner involvement is institutionalised and improvement decisions are made by consensus'. At Ratheon the implementation of QA and measurement was reported as being relatively easy because of the involvement of staff.

5.1.3 Creating a positive climate

A successful and effective programme will be viewed positively by all major staff groups within a company (senior managers, middle managers and developers). Initially generating and subsequently maintaining that positive climate is difficult. A number of reports recommend ways to tackle this issue:

- **A guru**
  Companies should get an external, eminent practitioner to initially 'sell' the idea of QA and measurement to practitioners (US Navy, Siemens, NCS). Contel had Rombach whilst Racal had experts from Hewlett Packard. This helps to initially generate a high level of credibility and seriousness to the implementation of QA and measurement.

- **A champion**

  A senior internal manager should be appointed to maintain the programme in the longer term (US Navy, Siemens, NCS).

Given the high rate of programme failure [Kearney 1992] it is important that companies put enough effort and resources into introducing and maintaining a programme effectively. This effort is reported as involving management commitment together with an investment of risk, time and cultural change (ICL, Siemens, US Navy, Motorola, Hughes Aircraft).

However at Hughes Aircraft and NCS they found that if QA and measurement is implemented progressively and positively then 'employee buy-in' can be achieved which results in a significant increase in employee company pride. This means getting the right balance between generating and maintaining enthusiasm for the programme and helping and supporting people to change, rather than ignoring the natural fear that people have about changing

Another good example of a company which has successfully created a positive improvement culture is the NASA/IBM's space shuttle team. Their CMM level five programme was developed on the basis of sound software engineering principles combined with an employee empowered culture dedicated to quality. The company successfully established a very disciplined QA and measurement-based development process focused on doing things right. However, this was supplemented by employees being empowered to continually question quality. As a result of this empowerment the programme achieved cultural acceptance within the company and became highly successful.

However, having described two very successful QA and measurement based improvement programmes, the published reports of both lack enough detail about *how* their cultural changes were brought about. Both companies seem to have been very successful at achieving a positive and empowered company ethos, but the subtleties of how that was achieved are not reported in enough detail. This, once again, highlights a weakness in what is actually reported and made public within the literature. On the other hand it might mean that neither company actually implemented, for example, employee empowerment in a true sense. Maybe employee empowerment for quality at NASA/IBM really meant individual practitioner responsibility and accountability for faults? Without more implementation detail it is hard to tell what this empowerment really meant.

5.2 Goal orientation

Having clear improvement goals is widely reported as an important indicator of an effective programme [Basili and Rombach 1988]. Companies realise that there has to be justification for the QA techniques and measures that are implemented, otherwise there will be no rationale for what is implemented. Consequently an increasing number of companies are goal orienting their programmes (for example, Siemens, Mitre and many more).

Specific company experiences cited in the literature include the following advice on goals:

Improvement goals should be:

- Clearly defined [Basili and Rombach 1988].
- Defined before the programme begins (Hewlett Packard, SEL, Racal).
- Based on the existing documented process (ITT Aerospace, NCS).
- Defined by the people actually working within the process, as it is only they who are in a position to define realistic and practical goals (Racal).

Companies should ensure that:

- Practitioners are aware of goals and are committed to them.
- Practitioners can see a clear link between what they are doing and how that is related to the achievement of goals (Contel, Siemens).

Clearly, there are major issues surrounding the initial selection of those goals, as, for example, studies have shown that developers can optimise a variety of goals [Weinberg & Schulman 1974]. It is, therefore, vital that the right goals are identified and thereby worked towards.

5.3 Communication and information

An effective programme is one where all staff groups are committed to the goals of that programme; ownership of the programme is a cornerstone of success. Communication and information coupled with participation are the tools by which companies can secure this ownership and commitment.

Companies which do not implement effective communication channels within their programmes are not only likely to have unsuccessful programmes, but they are likely to actually lower morale in the company. Indeed not getting communication right during an organisational change will be counterproductive.

Motorola found that securing consensus and acceptance of their programme by the user community proved to be crucial to success. NCS instituted a policy of always informing people about what was happening and agreeing with them what would be done and how it would be done. NCS also report that once such a policy of keeping people informed is in place it is then vital that the company actually does what it says it is going to do.

5.3.1 Raising awareness

All companies who have introduced QA and measurement stress the importance of raising awareness levels. They say that this should occur at all levels within the company, and should accommodate sharing QA and measurement-related experiences within the company.

A very effective way of raising awareness is by implementing mechanisms to spread success stories around the company (Contel, US Army, Siemens). All too often it is stories of QA and measurement disasters (usually apocryphal) which spread and do great damage to the attitudes of practitioners towards QA and measurement.

Racal is a good example of a company that instituted a highly successful awareness campaign. This was a massive attempt to educate staff at all levels of the value of QA and measurement. The campaign included training, newsletters, an internal day-to-day help desk, regular reviews and working groups. Indeed many of the companies with the most successful programmes report awareness campaigns as comprehensive as the one at Racal.

5.3.2 Information dissemination and feedback

Once a programme has been established then daily information and feedback is an important way of contributing to its sustenance and maintenance. Feedback, in particular, is reported as critical to success. Many companies report on the importance of providing feedback and results to all staff, and not just to managers (US Army, US Navy). This point is further confirmed by Cougar and Zawacki's work into motivating computer personnel [Cougar & Zawacki 1980]. In their studies they found that not only were feedback levels in the computing industry lower than in other industries, but also that personnel within the industry had a higher need for feedback than professionals in other industries. They did not speculate on why this was the case.

Feedback must also be a two-way thing as, for the programme to improve, it must be constantly refined and tuned on the basis of past behaviour and performance.

Communication mechanisms that companies have used to provide feedback and information include* :

- Newsletters (Racal, NCS, EDS General Motors)
- Internal conferences (Hewlett Packard, NCS, EDS General Motors)
- Practitioner guides (EDS General Motors)
- Regular reviews (Racal)
- A day-to-day support desk (Racal).
- A QA and measurement group/council (Hewlett Packard, Motorola, Racal, EDS General Motors, ITT Aerospace)
- A QA and measurement library/warehouse (EDS General Motors)
- The design of an *implementation template* for use by managers (EDS General Motors).

However, [Pfleeger 1993] also makes an important point about the nature of feedback and information. She says that it is important that the process and not the people are criticised in any feedback. Criticising people in this way will be counter-productive and alienating. This may result in an overt or covert lack of co-operation.

5.3.3 Training

All companies report on the need to train staff in QA and measurement. Various levels of training are recommended: overview level training to raise awareness and encourage participation, and detailed low level training in specific QA and measurement techniques (Contel, Hewlett Packard, Motorola, ICL, US Navy, US Army, ITT Aerospace).

5.4 The implementation process

In this section I discuss particular QA and measurement implementation factors reported in the literature.

How programmes are initially implemented is crucial to success. Too often the initial introduction of programmes is poor and results in ineffective or failed efforts. [Goodman 1992] advocates a 'project' approach to implementation. He believes that programmes should be implemented using the same principles by which new software systems are implemented. This means that companies should adopt a lifecycle approach to developing a programme and then introduce the programme into the user community with the same care, skill and sensitivity that would be used when introducing a new software

* Although some of these have been recommended as either QA or measurement specific mechanisms, they are almost all relevant to the implementation either.

system. This approach has merit as often the implementation principles that companies use to develop and implement software systems are not applied to non-software systems.

5.4.1 Planning

The implementation process needs to be well thought out and planned. Too many programmes have not been planned thoroughly and are implemented on an *ad hoc* basis. There are various aspects of planning a programme that companies need to consider, including:

- **Do background research**
  Doing proper background research is reported as an important aspect of planning the implementation of QA and measurement (Hewlett Packard, US Army, Siemens). Indeed, NASA/IBM say that one of their major success factors was actually putting into practice practical and innovative ideas from industry and academia. However, often companies do not find out what other companies are doing, what kind of techniques exist etc. Many companies end up repeating the same mistakes that are already publicly documented.

- **Analyse needs**
  Companies report on the importance of planning QA and measurement around the existing development process (ITT Aerospace, Contel). QA and measurement should minimally disrupt the existing process. This means that in the process of analysing for QA and measurement the development process should be understood and documented (Contel). Coupled with this advice is emphasis on the importance of communicating with practitioners and involving them in the development of QA and measurement. The practitioners who will be using the techniques should play a central part in their development. Management should just review rather than write plans (ITT Aerospace). The problem of motivating staff should also be addressed at this stage.

- **Tailor plans**
  It is widely reported that, because QA and measurement need to be harmonised with the existing process and calibrated to specific user environments (NCC), a number of different QA and measurement plans will be needed. Effective programmes are tightly scheduled and well integrated into both the engineering and management process [Card 1988]. Consequently, departments, teams and projects with different processes (or process maturity) will need different QA and measurement plans (Contel). Some companies make the mistake of trying to adjust and standardise processes to fit QA and measurement, but this is regarded by many commentators as a mistake. Plans should also identify the use of tools and be properly costed (Boeing).

- **Formalise plans**

  A detailed implementation plan must actually be created before a programme is introduced (NCS). Practitioners should be allowed to develop their own plans, as only they know what will work in their environment (ITT Aerospace). The plan must satisfy all information needs, at all hierarchical levels of the company and a successful implementation plan will have something in it for everyone (including some short term benefits for managers [Goodman 1995]). It must also be seen to address real problems (EDS General Motors, Motorola, ITT Aerospace, US Army). Feedback to developers must also be built into the plan (US Army, Motorola) and the plan should minimise extra work and be capable of evolving as the process evolves.

- **Piloting**

  Although there is a certain amount of dissent in the literature about the appropriateness of piloting QA and measurement, on balance there is more support for it than against it. Many companies encourage the use of pilots (for example, Contel), while only a few commentators are against it. The most notable anti-piloting advocate is [Goodman 1995]. He believes that piloting takes too long to produce results and that this time-lapse drains away enthusiasm.

5.4.2 An incremental approach

The advice that is most often published by companies is to start QA and measurement off small and simple and let it evolve slowly and iteratively (Motorola, NCS, ITT Aerospace, Mitre, US Navy, NASA/IBM). Commentators say that programmes like this not only allow practitioners to understand the relationship between what they are doing and why they are doing it, but also are less likely to overburden them. Programmes implemented in this way also allow a more 'seamless integration' of QA and measurement into the development process [Hetzel 1993], which will disrupt existing working practices least [Fenton 1991].

Many companies start with expectations that are too high and want very quick results. As a result they implement overly ambitious programmes (I observed this problem in my research).

[DeMarco 1982] suggests an effective approach to the initial implementation of measurement which also seems equally relevant to the implementation of any QA technique:

1. Select projects near their beginning and which are to be completed within 9-15 months, with about 15 staff working on them.
2. Expect a bit of chaos at the beginning.
3. Insist the group monitors the programme's performance and projections.

4. Evaluate the results of the programme.

DeMarco's advice is to select projects that are most desperately in need of process or product improvement, with project teams who are willing to participate, and projects that have good visibility within the company. When coupled with Shari Lawrence Pfleeger's advice [Pfleeger 1993] this seems a very effective way to get off to a good start. Pfleeger's rationale is that if you start with projects that are in need of most help, significant improvements are possible. The desperate state that such projects are in also means that instant converts will also be made of project staff. This may then generate positive internal publicity and success stories.

Having said all of this Boeing proved that it is possible to implement an apparently successful programme without taking much notice of any of these factors. In Boeing the programme was senior management dictated (the reason given: to make it useful, uniform, simple and unobtrusive to developers). The programme was also implemented in 3 months.

5.5 Tool support

Many reports urge companies to use tool support for QA and measurement, especially for the collection and analysis of data (Contel, Motorola, US Army, NCC, ITT Aerospace). Furthermore, companies are recommended to try and implement tools that are as simple as possible [Hetzel 1993, Baker 1991, Rombach 1990].

However, [Kitchenham & McDermid 1986] warn companies that although tools are useful, they do not always deal with some of the most important issues.

5.6 Resources

QA and measurement programmes are time consuming, costly and difficult to implement effectively without adequate funding. Yet, underfunding is a feature of many programmes. This usually results from lack of management commitment.

Although there is a consensus on the need for some kind of dedicated QA and measurement team there is no clear agreement on the extent to which this should be centralised as opposed to devolved, nor whether QA and measurement should be the sole activity of this team. [Grady & Caswell 1987] and [DeMarco 1982] recommend a centralised metrics group (like Hewlett Packard's metrics council) with DeMarco further emphasising the need to staff this group with very technically sound people. However, other commentators disagree with this level of centralisation (for example, [Pfleeger 1993]), believing that the programme should be very close to practitioners. This school of thought believes that proper ownership of the programme is easier to achieve without dedicated staff. This view also supports the premise that effective measurement must be an integral part of the development process [Baker 1991].

In analysing the published experiences there seems to be more of a case for dedicated QA staff than for measurement staff.

5.7 Evaluation

In keeping with Paul Goodman's 'project' view of implementation [Goodman 1992], many commentators recommend that companies rigorously assess, track and evaluate the affect of programmes on their products and processes (for example, [DeMarco 1982]). Many companies fail to do this effectively, and consequently very little data is reported on the real impact of implementing QA techniques.

[Jeffery & Berry 1993] have done some excellent work in designing a model by which a measurement programme can be assessed for effectiveness. Their model is as relevant to QA programmes as it is to measurement programmes.

They identify four areas of assessment which are loosely based on Fenton's approach:

1. **Context**. The environment in which the programme is developed and operated. This must have clearly stated objectives and goals and realistic payback assessments.

2. **Inputs.** The resources applied to the programme. Companies must resource the programme properly and, in particular, allocate resources to training.

3. **Process.** The methods used to develop, implement and maintain the programme. Company objectives must determine the measures, an independent measurement team must be in place, a database must be created, tools must be used, individuals must not be assessed and data must only be used for pre-determined purposes.

4. **Products.** Measures taken, reports produced etc. These must facilitate actions to be taken on the basis of observed measurements.

Programmes are assessed in terms of the criteria specified in each of the four areas of the model and predictions can then be made about how likely programmes are to succeed.

5.8 Factors specific to implementing measurement

5.8.1 The human issues

The way people react to measurement makes managing the human side of measurement critical. If a measurement programme is not carefully planned and sensitively implemented it can be counterproductive, as people may behave irrationally. This is because measuring will disturb the system being measured and staff will start to behave differently, as measurement is interpreted as setting a goal.

The following is a good example of measurement leading to irrational behaviour:

> *"Hitachi software in the 70's discovered a relationship between defects discovered early on and the number yet to discover (latent defects). So it tracked defects and on this basis projected latent defects. Consequently, programmers were put under pressure to find more defects if a lot were discovered early on. As a result of this programmers stopped declaring defects when they found them and saved them up to declare them much later the in process"* [DeMarco 1995, page 20]

In an attempt the try and minimise the potential damage that measurement can do, the following advice is cited as essential:

- **Collect data which relates only to products and process**
  Do not collect data on the development staff. People are very keen to criticise "the system", but they are not keen to have their own deficiencies pointed out (Contel). Although it is very tempting for management to use data to assess and punish people, this must be resisted (this is a particularly difficult issue to address when some companies implement a measurement programme primarily to track productivity). The data collected must be (and understood to be) thoroughly non-threatening to individuals (Hewlett Packard, NCS). Furthermore, measurement data must only be used for pre-determined purposes [Fenton 1991].

- **Do not ignore the fear that people have of measurement**
  Show them what measurement can do for them as individuals. People generally want to know how they, and the company, are performing (NCS). They also, invariably, want to see things improve (ITT Aerospace). Unfortunately, even some of the industry's most eminent commentators seem not to have understood the criticality of this factor (for example, [Card 1988], and the paradoxical advice from NCS to never use measurement for discipline, but to keep all data on individuals private).

5.8.2 Selecting measures

5.8.2.1 Principles of best practice

The following advice for selecting measures is most frequently quoted in the literature:

- **Initial measures must be simple and meaningful**
  Individual measures must be simple, precisely defined and objective (Kodak, Motorola). The collection of measurement data must also become an integral part of the software process and be unobtrusive enough not to disrupt existing working practices [Baker 1991]. The initial measures selected must also be closely coupled to a project's process maturity, the collection of data for which must be cost effective (Motorola). It is also suggested that programmes should begin by implementing measures that can be used as success or problem indicators, such as cost, schedule and quality measures (ITT Aerospace, US Army). Several companies, including NCS, suggest that programmes should begin by collecting data that is already readily available in the system (however, this is inconsistent with a goal-driven approach to measurement).

- **Measures must be thoroughly defined**
  Misleading data is collected if definitions are not accurate or well understood (Mitre, NCC). This is often difficult, for example, Racal had difficulty defining measures to a standard that was acceptable to everyone. NCS implemented measures that were too general and subjective and consequently inconsistent data was collected with which nothing could be done - it just satisfied a contractual obligation to collect data. In Boeing very simple definitions were found to work best. There must also be consensus amongst the people who will be collecting and using the measure, as to that measure's definition (Motorola).

- **Use several measures**
  No single measure will suffice, as a single measure can easily be manipulated (Kodak, Motorola). This is as a result of practitioners being able to maximise any one goal and means that it is important to measure progress on several levels. Consequently many reports urge companies not to rely on just one metric, and also to take into account other sources of information (Contel, NCS, US Navy, Kodak, [DeMarco 1982]). To address this problem [Fenton 1991 page 42] advocates a three strand process of identifying process, product and resource measures.

- **Don't collect too much data**
  Where lots of data is collected without a clear understanding of why it is being collected, data will not be used and will create a *data cemetery* [Fuchs 1995]. Furthermore, collecting too much data is said to

encourage casual, rather than serious analysis and results in a very weak measurement programme [Baker 1991].

5.8.2.2 A base set of measures

Although many publications offer a base set of measures to start off a measurement programme (for example, [Pulford 1996]), such a base set is difficult to justify, as different companies and different projects may have different goals. However, Hewlett Packard suggest the following initial set of criteria which measurement should address:

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

Hewlett Packard uses a variety of measures to address these criteria, ranging from straight-forward statement counts and payroll data, to the more sophisticated use of, for example, questionnaires designed to numerically assess system difficulty and determine the communication overheads associated with particular projects. [Cox 1991] gives more information on the measures Hewlett Packard uses and describes how the measures have evolved over time.

Finally, in selecting measures [Pfleeger 1993] says that using some measurement is better than using no measurement and that the biggest hurdle is to use any measurement at all. She believes that once the principle of measurement is established companies can then put effort into deciding which are the best measures to implement in a specific development environment. This belief makes intuitive sense, but contradicts all other advice on the relationship between choosing relevant measures and the ultimate success of a programme.

5.8.3 Data collection

When data is actually collected the following issues are reported in the literature as particularly important:

- **Data accuracy**

Collecting valid data is emphasised in the literature as fundamental to programme success. Indeed most companies agree that planning and implementing data validation is crucial (Hewlett Packard, US Army, NASA/SEL). There are several elements to data validity and these include:

◊ *Representation*: That data correctly measures the attribute being measured.

◊ *Accuracy:* That the data collected is accurately recorded (i.e. no counting mistakes or typographical errors have been made). Proper data controls should ensure data integrity and tracability of data as should the timely analysis of data (ICL). Indeed the US Army found that inadequate and out of date data with faulty or invalid analysis was a major problem.

◊ *Manipulation:* Whether the data collected has been 'fiddled' by the person collecting or analysing the data. [DeMarco 1982] says that in order for this to be avoided, data must be independent of the influence of project personnel and there must be no way to manipulate the data.

However, despite having data validation controls in place, in ICL they found that when projects were under pressure, there was a tendency to just be seen to be recording data and not to worry too much about the accuracy of it

- **The black hole**

It is essential that the data collected does not disappear into a black hole. Equally important is that data is not <u>seen</u> to disappear into a black hole. Thus, feedback on the data collected must be provided to the practitioners who collected the data. If this does not happen then motivation to collect good data will disappear and bad data will be collected (NCC, US Army, NCS, US Navy, [Hetzel 1993]). Indeed at NCS all data is made public. Furthermore a programme that has good feedback mechanisms can get the benefits offered by the "experience factory" [Basili 1995].

- **Data Presentation**

All companies agree that data must be presented in an easy to use way and at appropriate levels of detail for different audiences [Hetzel 1993]. In Boeing, for example, simple data is presented graphically, and this data is frequently used and widely distributed (in this research I found that programmes where data was frequently used were most likely to be successful). Presentation of data is seen as particularly important in US companies (for example, Mitre), where inappropriate presentation (too detailed or too vague) was found to be very damaging (for example, US Army). In this research I found the presentation of data was less important in UK companies.

5.8.4 Tools for automated data collection

Almost all of the companies reviewed emphasised the need to automate as much data collection and analysis as possible. Indeed the provision of a basic set of tools was viewed as very important. The sophistication of the tools implemented varied enormously between companies. Spreadsheets proved to be the most popular analysis tools (for example, Contel, Hewlett Packard, Boeing), probably because most people were familiar with it.

However, opinion was divided about the storage of measurement data. Some companies and commentators favour the introduction of a company-wide measurement database or 'data warehouse' (Hewlett Packard, EDS General Motors, [Card 1988]), whilst others favour small locally based databases (Motorola). However, the emerging consensus tends towards the latter initially, and the subsequent development of a larger scale, company-wide database when the measurement programme is more mature. What is not disputed, however, is that whatever storage medium is used, practitioners must have access to data (Contel).

6. Conclusions

I have shown that the large body of literature on implementing software QA and measurement programmes has two major weaknesses:

1. It is too skewed towards describing the tools and techniques to implement, rather than the process by which to implement them - there is too much *what* and not enough *how*.

2. Very little of the long established and validated generic implementation theory that exists in the social and management science literature has been referred to in the software implementation literature.

Much of the literature neglects the most difficult and critical human issues of implementation, or else pays lip service to them. Indeed the literature sometimes makes implementing QA and measurement sound easy.

Furthermore, although there is a large body of literature on QA and measurement generally, only a small sub-set of this is software development specific, and, furthermore, even less of this is tangible and easy to implement advice. Overall, the QA and measurement literature suffers from being generally too 'wishey washy'. Quality is talked about in general terms that are not very helpful to software practitioners and, furthermore, there is little consensus as to what constitutes effective QA and measurement mechanisms. For example, a particular school of thought urges companies to have zero tolerance of faults, but does not really explain how this can be achieved and what steps companies can take when faults persist. On the whole the literature revealed that QA and measurement in software development is not mature, well defined nor particularly pragmatic.

Chapter Four

The currently reported state of practice and recommendations

1. Introduction

In this chapter I present an overview of the best QA measurement programmes as reported in the literature. I examine many published accounts of companies' QA and measurement experiences, and although I am most interested in *how* the companies have gone about implementing their programmes, I do also report on *what* some of them have implemented.

The reports published in the literature are best practice reports of, usually, large successful companies (most of which have self-selected into the literature). In Chapters Five to Eight I use this chapter as a baseline to compare reported best practice with what was actually practised in the companies in my study. I found that the practice I report here is not common throughout the software industry. Furthermore, I also found that the published state-of-the-art (as presented in the previous chapter) is heavily influenced by the best practice I report in this chapter, especially the practice from, for example, Hewlett Packard and Motorola.

Despite most published accounts reporting very positive outcomes of QA and measurement I found that the problems companies did identify were related to weak change management. Many companies reporting basically successful programmes, in explaining the development of the programme, recounted basic change management problems (for example, Contel). Furthermore the few unsuccessful programmes reported were directly hampered by poor change management (for example the unsuccessful programme reported by [Jeffery & Berry 1993]). Reading between the lines in these published reports it is my view that many companies were not knowledgeable and skilful in change management and some were naive enough not to really understand that there was even an issue to address.

Many of the reported companies appeared not to understand the interdependence of change and the practitioners affected by that change. Furthermore, such companies failed to appreciate how counterproductive a poorly implemented change can be. Indeed, many accounts make implementing successful QA and measurement sound very easy. It is these companies that make the mistake of concentrating exclusively on *what* to implement, neglecting entirely *how* to implement it.

The chapter is structured as follows: In Section Two I discuss what has motivated the companies reported in the literature to introduce QA and measurement. In Section Three I discuss the costs and benefits of QA and measurement reported by companies. In Section Four I analyse the reasons why many companies fail to implement state-of-the-art programmes. Section Five presents a summary of the reported company case studies that I have drawn most heavily on. In Section Six, I identify the main QA and measurement problems that have been encountered by the companies reported in the literature. I present our conclusions in Section Seven.

2. The motivations for companies to introduce QA and measurement

Although software companies cite a variety of reasons for introducing QA and measurement, almost all of these can be broadly categorised as follows.

2.1 Product control

Some companies initial motivation was related to tracking specific software product attributes. Tracking and improving defect densities was most common. For example, NCS wanted to fulfil a contractual obligation to track defects. Other companies had wider product interests, for example, SEL's initial aim was to be able to objectively profile and compare projects.

2.2 Process control

Many companies realise that a lack of process formality means that project decisions are largely subjective. Consequently many companies implement QA and measurement, to:

- adopt standard approaches to software development;
- evaluate new and existing tools, techniques and methodologies.

Most of the companies I reviewed realised that they knew very little about the software that they were producing, little about quality levels or about the low level costs associated with particular development processes. Obviously companies are universally concerned about product quality, cost effectiveness and productivity. Boeing, for example, wanted to monitor their outsourcing suppliers. Managing these two factors effectively, has led to companies identifying a need for quantified data, *i.e.* measurement.

2.3 Process improvement

The move towards process improvement and assessment, in particular the increased popularity of the SEI Capability Maturity Model (CMM) [Humphrey 1989] has also significantly increased companies' interest in QA and measurement techniques; not only has the movement raised awareness of QA and measurement, but there is an implicit need to incorporate more sophisticated QA techniques, including measurement into the development process to reach the higher levels of the model (for example, EDS General Motors, Racal). Moreover, there now appears to be consensus on the need for measurement within improvement programmes [Pulford 1996, Pfleeger & McGowan 1990]. Indeed almost all of the companies I reviewed cite process improvement as one of the most important reasons for implementing QA and measurement. However given that most companies are at a low level of process maturity, there did seem to be more interest in measurement than might be expected when this is only really appropriate at the higher levels of CMM.

Many companies, whilst not formally assessing themselves against CMM are interested in improving their development process. Whether formal models are adopted or not, companies realise that if they want to improve their development process then it must have, in the first instance, a quantified and objective benchmark (for example, NASA/SEL, Kodak). This, again, calls for the introduction of software measurement.

Although Hewlett Packard were probably the first to cite measurement as a valuable tool for improving the development process, many companies have since recognised the relationship between measurement and process improvement. For example:

- SEL implemented measurement in order to "provide discipline within project development";
- the US Army were keen to develop measures which would provide management information to help predict the behaviour of software during its development, with the ultimate aim of minimising cost overruns;
- Racal were concerned with improving the accuracy of their project estimates;
- Kodak wanted to be able to provide management planning information for estimating and predicting project behaviour and for identifying projects for re-engineering;
- Boeing wanted a simple and common basis of communication across programmes.

2.4 Certification

Although, as a general trend, companies are increasingly moving towards quality certification, relatively few of the accounts I analysed mentioned certification as a direct motivation for implementing QA and measurement (an exception was Boeing who were aiming for a high CMM assessment). This is surprising as, whilst advanced QA techniques like measurement are not explicitly required for certification, the rigorous application of a certified quality system, such as any of the ISO9001 series, would really demand, for example, measurement be in place before certification. Interestingly two companies were already ISO9001 certified before implementing measurement [Bazzana *et al* 1995].

2.5 Senior management dictat

Some of the companies I reviewed had no stated objectives for their programmes. EDS General Motors, for example, introduced measurement because that is what "the management" wanted. Indeed senior management dictat was a relatively common 'motivation'.

3. Benefits and costs of QA and measurement

3.1 Benefits

Unfortunately, companies which have introduced programmes rarely quantify the benefits derived from this. This lack of quantification is, unfortunately, a problem regularly encountered in software engineering [Hetzel 1995] and means that I am forced to report the subjective benefits that companies cite publicly. It may be the case that companies do measure benefit but are reluctant to publish that data. However, in my research I found no evidence of this.

Companies report many subjective benefits from implementing programmes, most of which are related to improved process control and management.. Many companies are forced to think explicitly about their existing development process for the first time when they come to plan the implementation of QA and measurement. As a result of this, improved process understanding is reported as very valuable (Hewlett Packard, Contel). Furthermore, the precursor of understanding the process makes that process more visible and transparent within companies (Siemens). This improved focus on the process (rather than simply on the product) means that process improvements are more effective (Contel, Hewlett Packard). Furthermore, [Kitchenham 1993] views measurement as an important risk reduction strategy.

Some reports (Contel, Motorola) also stress that QA and measurement has the indirect benefit of highlighting and making company goals explicit to practitioners. Furthermore progress towards improvement is also made more visible using measurement.

Other significant benefits that companies have derived from implementing QA and measurement include:

- Improved communications and in particular proper discussion of issues, progress and problems (Hewlett Packard, Boeing)
- The objectivity of process management (Hewlett Packard, US Army, Contel, Siemens)
- Improved product quality (BT, ITT Aerospace, Motorola, Siemens)
- Increased programmer productivity and/or efficiency (BT, NCS, ITT Aerospace, Siemens)
- Improved programmer work pride (NCS)
- More effective deployment of personnel and resources (Boeing)

The highly successful process improvement programme at Hughes Aircraft is a good illustration of the benefits that are possible with QA and measurement (this is discussed in more detail in Section Five). Hughes implemented comprehensive QA and measurement in its improvement programme and quickly succeeded in moving from CMM level two to level three. The key benefits from higher maturity that are of direct relevance to this thesis include:

- The uniform implementation of processes
- The uniform implementation of QA techniques (for example, reviews) across all projects.
- An improvement in the quality of QA staff.

The company also experienced some major side benefits that it was not expecting, including: higher morale, better performance and reduced costs. The company also found that its increased maturity decreased the risk of missing cost and schedule estimates and it has had no major crisis since the improvements.

However, QA and measurement are only tools and a means to an end. Measuring things has no merit in itself, it is how companies change their behaviour as a result of the data collected that actually improves quality (or whatever else a company is trying to improve). Consequently the literature urges companies not to over emphasise the *collection* of measurement data as a direct benefit.

3.2 Costs

There is very little data about the cost of QA and measurement in the public domain (especially QA). Many companies do not measure the costs of QA and measurement, and the few that do refuse to disclose their data. However, it seems likely that costs are measured more often and more directly than benefits. Companies are most interested in costs, and costs are easier to measure and are more measurable in the short term.

It is believed that programmes cost somewhere between 4-6 % of a project's budget [Rombach 1990, Kettler 1993]. It will also take about a year to produce enough measurement data that is useful [DeMarco 1982], but up to three years to generate enough data to be able to accurately predict trends [Grady & Caswell 1987]. [Goodman 1995], however, believes that it is possible to implement something of direct value in 6 months and his view is supported by Humphrey's report of the Hughes Aircraft improvement initiative [Humphrey *et al* 1991], where significant improvements were implemented relatively quickly.

4. Impediments to implementing the state-of-the-art

In reviewing the software literature it is clear that QA and measurement practice has not progressed very quickly. Indeed, many of the issues reported ten years ago continue to be pertinent today (for example, the issues discussed within ICL's measurement programme [Kitchenham & McDermid 1986]). However, over the whole industry there does seem to be three issues that are very relevant to any discussion of QA and measurement:

4.1 Poor software engineering management

Any analysis of the literature quickly reveals how neglected management and organisational issues are throughout the software industry. Indeed other studies report very poor software management generally [Jones 1994, Bache & Neil 1995, Keinayel *et al* 1991, Cougar & Zawacki 1980]. The poor management of QA and measurement that I found is not, therefore, unique as it seems likely that most things are managed poorly in the software industry. Furthermore, it seems that personnel are motivated by different things compared to personnel in other industries [Keinayel *et al* 1991], and that makes skilful management even more important. However, most published accounts of QA and measurement reveal much more interest in the technical, rather than organisational and managerial aspects of QA and measurement. Indeed, there seems to be a general attitude within the industry that implementing any new tool, method or technique is a management not technical problem and is, therefore, someone else's problem (for example, [Kitchenham 1984]).

[Bache & Neil 1995] give a good example of poor management within the industry:

> *"A QA manager said that a static analyser that checked for structured programming was not necessary because structured programming was always used (referring us to various standards and procedures that were in use). He was then surprised to be told that highly unstructured code was actually being written. He then blamed one very junior programmer for the situation and argued that structure was not that important anyway."*

This example not only illustrates poor management, but also shows how difficult improvement is when managers will not recognise how bad things currently are - a phenomenon that I also observed in my research.

In Japan good basic management and its relationship to quality improvement is well understood. Indeed, [Iizuka 1995] argues that the bulk of software faults are due to poor management rather than to poor technology. He also emphasises the importance of management by facts rather than by experience or 'gut feeling'.

This view is further confirmed by the NASA/IBM space shuttle software development team, which attributes its success to *"sound program management"* [Billings *et al* 1994]. Their CMM level five status was not based on implementing sophisticated technology; their success was based on organisational, procedural and cultural innovation. Something, they say, which can be implemented by any company with a real desire to improve.

4.2 Top-down implementation

All of the published accounts of the implementation of programmes are basically top-down. Indeed getting management commitment is emphasised above many other things. This approach to implementing change is counter to much organisational theory which advocates bottom-up approaches to change where possible. This might be a symptom of the software industry again thinking that it is special and believing that things that work in other industries are not appropriate to the software industry. Alternatively, it may be a symptom of companies within the industry organising themselves along very conventionally rigid and hierarchical structures, thereby making top-down implementation of change the only viable option. If this is the case then the many disadvantages of this form of organisation will probably soon manifest themselves in the companies which have implemented QA and measurement in this way.

However, Hewlett Packard seems to have taken most notice of organisational theory and claims to have implemented measurement in a bottom-up way. Indeed Hewlett Packard's programme was inspired by Japanese ways of working. It is, however, difficult to see, from the published accounts of Hewlett Packard's programme, how it is bottom-up. [Hetzel 1993] is one of the few QA and measurement commentators who really seems to understand the reality of bottom-up and top-down implementation and argues for bottom-up approaches.

4.3 Other reasons for companies not implementing QA and measurement

Many companies do not implement programmes, and commentators speculate on why this is the case. The reasons that [Kitchenham 1993] and [Card 1988] suggest include:

- Staff would not like it.
- It won't solve my problems.
- Don't know what to implement.
- All projects are different, so how can a uniform process be introduced.
- Have never had to develop software in that way before.
- It is too expensive.
- QA and measurement methods too immature to use.
- Too many competing unsubstantiated single metrics around.
- Incompatible to the way things are currently done.
- Managers fear arbitrary evaluation.
- Developers are too busy already.

5. A summary of the reported company studies

In this section I summarise some of the key company case studies that have been published in the literature. I summarise only a sub-set of those QA and measurement reports that have been published and that I have cited elsewhere in this thesis. Furthermore, it is worth noting that while many of the case studies I summarise report measurement programmes, these programmes are within a context of process improvement.

5.1 Boeing

[Lytz 1995] presents a case study of implementing measurement into the Boeing 777 project. The measurement programme is a top-down programme, having been instigated by senior management. Senior management wanted the programme to be primarily based around time series data which looked at planned versus actual data. The initial intention was to implement a useful, uniform, simple and unobtrusive set of measures into the development process within three months. A timescale that Lytz reports was met. Before this measurement programme a number of informal measurement efforts were in place, as different departments and teams realised the usefulness of collecting and analysing data. Consequently the aim of the company-wide programme was to standardise these measurement efforts so as to create a common basis of communication. The implementation plan for the measurement process was to start very small and simple and from that basis gradually develop the programme. Furthermore the implementation plan also depended on the use of tools, in particular the use of simple spreadsheets.

The measurement process is now well established and, Lytz reports, successful. He describes the programme as both tightly scheduled and close to the organisation. He also says that the measurement data is widely distributed and regularly used in decision-making, particularly in resourcing decisions.

He lists the following lessons learned from implementing a measurement programme:

- Often the discussion that data promotes is more useful than the data itself.
- The data being seen to be used be senior management was important.
- Implementing measurement in the middle of projects worked successfully.
- Simply defined measures were most popular.
- Simple to use tools like spreadsheets were most popular.

5.2 Contel

In Pfleeger's report of Contel's measurement programme [Pfleeger 1993] process understanding is emphasised as a key programme success criteria. Furthermore, matching the maturity of measurement with the maturity of the development process proved to be very successful at Contel. This principle was used to implement different measurement programmes for different projects - depending on the level of maturity of the development process of the particular project. Pfleeger also reports that piloting played an important role in the successful implementation of measurement. Pilot projects were carefully selected and then closely monitored, successful pilots were then widely advertised within the company. Pfleeger also reports that the measurement effort cost 5-10% of a project's effort.

The lessons learned at Contel about measurement are reported as:

1. **Begin with the process.** This especially helped to overcome developers' initial hostility as data is collected on the process and not them. Furthermore, at Contel developers were very keen to collect data 'on the system'.

2. **Keep the metrics close to the developers.** There were no separate metrics people at Contel and they found that this helped. Open access and instant feedback to data also help create a participative and non-threatening atmosphere about measurement.

3. **Start with the people who need help.** Contel adopted a policy of starting measurement in projects that were in trouble, that way the people in the project were very keen for anything that might help them, but it also meant that big results were possible. This policy was successful in not only creating instant converts to measurement, but also in generating lots of internal publicity.

4. **Automate as much as possible.** Contel's attitude was that effort should go into reviewing and analysing data, not collecting and tallying it.

5. **Keep things simple.** It was Contel's belief that developers should not have to become measurement experts to be effective within the programme, but also that the programme would be more successful if developers could see a clear relationship between what they were collecting and why they were collecting it.

6. **If developers don't want to collect particular data then don't make them.** Contel's position was that to make developers collect data they were unhappy collecting would just invite inaccuracy and incompleteness.

7. **Using some metrics is better than using no metrics.** Contel found that the biggest hurdle was to use any, so once a few are established then the number can be increased with much less risk.

8. **Criticise the process not the people.** All the measurement data and the consequent actions should be in process terms.

5.3 EDS/General Motors

[Kettler 1993] reports on implementing measurement in a very large organisation. Again the programme formalised the existing *ad hoc* informal and unofficial measurement efforts that were in place throughout the company. These informal efforts had no operational definitions nor consistent collection or analysis. The formalisation of measurement into a measurement programme was done via an 'edict from on high' and the process included setting up:

- A measurement task force.
- A library of measures.
- Structured training.

The aim was to implement measurement as part of an overall process improvement initiative. Again the programme was very much top-down, but had an inter-disciplinary team leading the initiative. Again this measurement programme is reported as being successful.

5.4 Hewlett Packard (HP)

[Grady & Caswell 1987] report on the influential measurement programme at Hewlett Packard. The programme is based on a measurement council of about 20 respected, nominated managers and engineers). This council:

1. distributes information;
2. proposes improvements;
3. attends meetings and shares results;
4. actively improves the process.

Hewlett Packard's popular ten point approach to implementing measurement is:

1. Define objectives for the programme.
2. Assign responsibility.
3. Do research.
4. Define initial measures to collect.
5. Sell the initial collection of those measures.
6. Get tools.
7. Training.

8. Publicise success and encourage the interchange of ideas.
9. Create a database.
10. Establish a mechanism for changing things.

In [Grady & Caswell 1987] (and its follow-up [Grady 1992]), many important, and usually neglected, organisational issues surrounding software measurement are identified.

5.5 Hughes Aircraft

[Humphrey *et al* 1991] reports on the very successful improvement programme at Hughes Aircraft. This is one of the few programmes that can be quantifiably and objectively assessed as successful, as the programme resulted in the company rising from CMM level two to level three in two years. In addition, the programme is also unusual in that financial information is available which shows the costs and benefits of the programme (though there is little information on how the financial costs and benefits were calculated). The programme cost $445,000 to set up and, so far, has been estimated to have saved $2 million.

Humphrey also reports that the other benefits observed by Hughes include better working conditions, improved morale, better performance and lower costs. Humphrey reports that some of the key improvements that were made to the process include:

- Implementing uniform definitions for measures.
- Resourcing proper data collection.
- Using a central database.
- Increasing the resources for the quality department (which led to additional and better trained staff).
- Implementing a uniform review process across the company.

The company reports that the benefits of the programme have far outweighed its cost.

Humphrey also cites the following lessons learned:

- Management commitment is crucial. This means management must invest resources, time, risk and implement cultural change.

- Improved company morale is a major, intangible, benefit. This also leads to even more 'buy-in' from staff, which, in turn, makes the programme even more effective.

- An increase in maturity resulted in the company experiencing less risk. Since the programme resource and timescale schedules have been met and the company has had few crisis of the type that led them to implement the programme.

5.6 ITT Aerospace

The measurement programme that [Karalak 1993] reports on is, again, a top-down initiative. However, the measurement process at this company was reviewed, rather than written, by senior management. In particular, the software engineers who would be collecting and using the data played a central role in identifying the process by which the measures would be collected. Furthermore, this programme is closely related to the development process and, because each project is likely to have a different process, the programme has been tailored to suit individual projects. Again, Karalak cites automatic tools as important to the success of measurement.

The lessons that ITT Aerospace say that they learned about implementing programmes include:

- Vision is needed.
- Senior management sets the tone.
- Measurement must be directly related to parallel improvement activities.
- Data must be useful to both engineers and managers.
- Trust and commitment is necessary.
- Most people want improvement.

5.7 Kodak

[Carlson & McNurlin 1992] report on the implementation of measurement at Kodak. Although this is another measurement programme primarily instigated by outsourcing, another key motivation was a need to identify projects for re-engineering. Furthermore Kodak also wanted measurement to help assess and improve processes. They particularly wanted to collect data in answer to the following questions:

- What are our key processes?
- How well are we doing them?
- How can we improve them?

Kodak's process goals also made them baseline, in particular they baselined quality and efficiency - they were probably one of the first companies to recognise the importance of baselining as they did this in 1988.

Kodak's guiding principles include:

- Analysing 'before' and 'after' implementing new methods or tools.
- Focusing equally on business and technology factors.
- Not relying on just one measure.
- Educating senior management on IT, and using measurement data to do this.
- Benchmarking both internally and externally.

5.8 Motorola

[Daskalanakis 1992] reports on the measurement programme that was implemented at Motorola. The programme here was an integral part of their well known 'six sigma' quality improvement initiative. This initiative aimed to deliver no more than 3.4 defects per million output units from a project. It tackled this product improvement by process improvement. Consequently the programme is very process orientated. Also, the extent to which cultural and human issues are addressed by the company make this programme quite unusual. Motorola was keen to get consensus and acceptance by the user community and also emphasise the need for the training and support of metrics users. However, initially the programme was senior management driven.

Motorola's programme is reported as being well researched and planned. The Hewlett Packard approach to measurement seems to have been implicitly used in this programme. Furthermore a GQM approach to identifying measures was also implemented. Very particular audiences for metrics were identified. These included: users, senior managers, software managers, software engineers. All of these different audiences were identified as having different needs from data. To this end, although a company-wide measurement infrastructure was implemented, project and process-specific measurement was implemented that was very local to the collectors and users of data.

The importance of implementing objective, cost effective and informative measures that are actually used is also emphasised by Fenick. Although Motorola did not initially implement a central database, instead relying on small local repositories, this has been changed as users wanted a central store.

5.9 National Computer Systems (NCS)

[Stevenson 1993] reports on implementing measurement at NCS, something they did as a contractual obligation (outsourcing makes this is an increasing motivation). The importance of the cultural aspects of measurement is emphasised in this account, in particular NCS found that creating and maintaining enthusiasm for measurement was crucial as was making measurement part of the organisational culture.

Once measurement was implemented in the company, the initial attempt is cited as weak. Problems inherent in the first programme included: measures implemented that were too general and subjective, they were not defined precisely enough; the data collected was inconsistent; the data was not used effectively and was just recorded to satisfy the contractual obligation. However, the programme has since been improved dramatically to the extent that senior management are very interested in the data and decisions are made on the basis of it. Furthermore, the usefulness of data has been realised by the development staff and the programme is reported as much more effective and successful.

5.10 Racal

[Shelley 1993] reports on the measurement programme at Racal. This is another programme which is very process improvement orientated. Racal first wanted to improve the estimating process, then the wider development process and product quality. Although this is another programme initially introduced by senior management dictat, they have recognised the importance of human and cultural issues. An 'awareness campaign' was instigated to first educate staff at all levels on the reasons for implementing measurement. Then more focused training was used to address how measurement would be implemented. Racal's programme also has many aspects of Hewlett Packard's approach, though, again this is not explicitly acknowledged.

The lessons learned by Racal include:

- The need for commitment from the top, as without this the programme is an easy target for resource cuts.
- The importance of understanding the development process before implementing measurement, and then implementing process-specific measurement.
- Simple measures and measurement processes work best.
- Maintaining a high level of communication is also important.

5.11 Software Engineering Laboratory (NASA/SEL)

[Valett & McGarry 1988] report on the implementation of measurement by SEL at NASA. They not only emphasise the need to baseline and measure change against this, but they also particularly emphasise the importance of a proper approach to data validation (three layers of data validation were introduced in NASA's programme). GQM was used to identify measures and a mixture of automated and form-based data collection methods were used to collect both objective and subjective measures. A central database was used to store the data collected.

This is one of the few programmes to have measured the cost of data collection. Measurement at NASA cost 2-3% of the total development cost (they had predicted a cost of 8-10%). However, benefits are not reported in such quantified terms. The benefits reported are all subjective but include:

- Improvements in quality and productivity.

- Providing information that management can use to plan projects more effectively. This included information that helped improve the estimating process, and which helped to predict project behaviour. The data collected also provided a baseline which was used to evaluate the impact of new technologies.

- Supporting process understanding. NASA used the data collected to develop profiles of types and causes of errors and then record the effort that was necessary to correct those errors. By doing this it was possible to identify particularly weak areas in the process.

- Providing project discipline. Data collection is reported as not only giving the developers a clearer understanding of the process, but by collecting data more discipline was introduced into the process.

5.12 US Army

[Fenick 1990] reports on the measurement programme implemented in the US Army. The aim of the programme was to improve an existing *ad hoc* programme, and in particular address the following problems with it:

- Data was inadequate and out of date.
- Faulty and invalid analysis was performed on the data.
- Data was presented inappropriately.

The programme was particularly geared to implementing preventative measures in the acquisition of software. In addition, senior managers stated a requirement for some short term pay offs - something increasingly being demanded.

Fenick cites the following lessons learned:

- Baselining is crucial.
- Do background research.
- Promote the programme and feedback on data.
- Make sure that there are some short term pay offs.
- Real needs must be addressed by the programme.
- Support and guidance must be provided to users.

5.13 US Navy

[McGarry 1996] reports on the US Navy's comprehensive approach to measurement. The programme was devised in response to late, cancelled, over budget systems that did not fulfil their user requirements, and which were primarily delivered by external suppliers (outsourcing is an increasingly cited reason for more formal approaches to QA and measurement). The main aim was to provide visibility and control throughout the lifecycle. Their approach aimed to provide a basis for the early detection and correction of faults and an objective basis of communication between managers and developers.

This is one of the few publications that acknowledges that actual and official processes may be different, but that data must be collected on the actual process. Furthermore, McGarry suggests that the onus is on managers to understand the developers' process and data (rather than the other way around).

The Navy proposes a flexible approach that can adapt to different needs, and one that also tries to address cultural change within the company. Like Pfleeger's view of measurement [Pfleeger 1995] the data collected is process specific. Concentrating on data that is normally collected by the developer is advocated, which is implicitly counter to GQM.

The main principles of this approach include:

- Collecting low level engineering data that is specific to and seen within the context of a particular project.
- Using data to support objective project management decision-making.
- Using a small set of simple measures that everyone can understand.
- Ensuring that proper commonly understood definitions of data are in place.

6. General problems

I found that many of the individual best practice programmes reported in the literature have weaknesses. Many appear to have been designed without reference to the state-of-the-art, as described in the previous chapter, organisational theory is especially neglected. Consequently, it is not surprising that some commentators report a programme failure rate as high as 80% [Kearney 1992] together with a generally high level of dissatisfaction with programmes [Hetzel 1993]. Indeed, it is reported [Card 1988] and widely believed that the state of practice is currently very poor. In this section I discuss the main reasons for the problems in the programmes that companies report.

6.1 Background research

I found that many of the reported programmes (even some of the better ones) seemed to be based on an inadequate use and understanding of the existing guidance on the effective implementation of QA and measurement. Indeed, contrary to much of the published advice (for example, as given by [Grady & Caswell 1987]), most companies do not report having done background research into QA and measurement. By relying solely on internally developed implementation strategies and using no external sources of guidance and information most companies seem to be re-inventing the wheel.

6.2 Organisational factors

Most companies do not take proper account of the complex organisational factors that are relevant when implementing new programmes. Indeed, I found some examples of outright antipathy from the software industry towards accounting for organisational factors. For example, White concluded:

> "*Successful change efforts focus on the work itself, not on abstractions like culture or participation*" [White 1993]

This apparent reluctance to acknowledge organisational issues even occurs in some accounts of successful implementation efforts. Exceptions to this include [Rombach 1990] and [Neilsen & Timmins 1997] both of which include a discussion of organisational factors.

6.3 Poor management

Poor management seems to be a major issue in some of the reported programmes. This is highly related to a failure to do any background research and failing to account for organisational factors. The following are the typical management problems that emerged from the published reports.

6.3.1 Unrealistically high expectations

Managers often have very high expectations that programmes are going to deliver quicker benefit, more easily than is likely. Consequently it seems common for managers to implement too much, too early.

6.3.2 Lack of focus

Managers seem to be bad at setting improvement goals effectively. To compound this it is also common for companies to implement changes that are not goal specific. This results in bureaucratic QA procedures and data that managers do not know what to do with.

6.3.3 No change management

This is a major problem. Managers tend to overburden practitioners with extra work and expect them to use cumbersome new procedures. Change momentum is often lost, and progress is not tracked. Furthermore, it is common for companies to provide no, or insufficient staff training.

6.4 Resistance to the programme

Surprisingly only Siemens really admitted to encountering any resistance. There developers claimed that the programme would be too costly, that it would entail extra work for them and that it would restrict their creativity. Developers at Siemens were also not convinced that there was a problem with their existing development process.

6.5 Measurement-related problems

An ineffective measurement programme can be very damaging. Indeed a poor programme was cited as one of the major causes of a major software development project failing at Paramax Electronic Systems. This was the result of inappropriate measures being collected and used as if they were useful.

Furthermore, managers at Paramax were reported to be too keen to accept pretty graphs as useful data, and were naively satisfied when they saw *"a neat array of lines moving towards a target like total size"*. Typical problems associated with measurement programmes are reported to include the following.

6.5.1 Selecting inappropriate measures

Companies must carefully select the measures they implement, as the attributes that are measured will appear to practitioners as very important. Effort will go into maximising those attributes [DeMarco 1982]. So if companies pay lip service to improving quality attributes, but only actually measure completion time, that is what most people will focus on (which is of course what most non-metricating companies do anyway) [DeMarco 1982]. It is, therefore crucial that the right measures are selected as practitioners can optimise almost any objective (rational or irrational). This phenomenon is aptly illustrated by Weinberg & Schulman's famous programming experiment [Weinberg & Schulman 1974].

Furthermore, [DeMarco 1982] says that where people are anti-measurement it is usually the case that those programmes are measuring the wrong things and that this is interpreted as resistance to measurement.

6.5.2 Mis-matching process maturity and measurement

Companies have significant problems if they do not couple the sophistication of measurement, to a particular process's level of maturity (Contel, Motorola). This means that, since most companies are at a low level of maturity, measurement implementation must not be too quick or too sophisticated. Indeed some commentators argue that companies should already have a reasonably mature process before implementing measurement [Baker 1991]. If a company has a chaotic process that is poorly understood then measurement is not going to help [Pfleeger & McGowan 1990].

Managers also often believe that measurement *per se* will improve the process of software production [Hetzel 1993].

6.5.3 Collecting invalid data

In ICL's measurement programme capturing inaccurate and misleading data proved to be a big problem. The following problems meant that invalid data was collected:

1. Practitioners sometimes misunderstood the measurement process.
2. Practitioners mis-recorded data, because:
    - overly complex data classifications had been designed by managers;
    - no was training provided;
    - they disliked being monitored;
    - the work overhead was considered too big.

Again these are probably common problems (and ones often reported anecdotally), but problems that are seldom reported in public.

6.5.4 Shortage of useful 'off-the-shelf' measures and methods

Another common problem that companies have experienced is to do with the actual measures used. Companies still complain that there are not enough:

1. well defined, and commonly used, measures (Motorola, Siemens);
2. rigorous guidelines for data collection, analysis and interpretation (Motorola, US Army);
3. automated tools for data collection (Motorola, Siemens).

This section is based on the work of: [Goodman 1995, Möller 1993, Rubin 1990]

7. Conclusions

Overall, I found that the state of the literature reflects the generally immature state of QA and measurement. There is not a clearly differentiated state-of-the-art and state-of-the-practice. Furthermore, the part of the literature purporting to represent the state-of-the-practice, actually reports the best practices of high profile companies like Hewlett Packard. There is also circularity in the literature: best practice is not always derived from the state-of-the-art, instead best practice seems to influence the state-of-the-art.

There are also weaknesses in what is actually published. In particular, most accounts seem to report only the marketable and sanitised aspects of companies' experiences. An illustration of this phenomenon is the very small number of documented experiences of practitioner resistance to the introduction of QA and measurement. Very few companies admit publicly to encountering resistance during the introduction of a measurement programme, but practitioner resistance is regularly reported anecdotally.

Furthermore, although lots of the case studies published emphasise the importance of a company-wide quality culture, most seem really to be paying lip service to this notion. Very few reports actually say what cultural issues need addressing and how they have been addressed. Instead fairly superficial mechanistic changes are described that do not represent cultural change.

Although QA and measurement seems popular at the moment, implementation by companies remains slow. Indeed, as with many new tools and techniques in the software industry many companies seem to be waiting to see what competitors do about QA and measurement before they commit themselves. However, in this chapter I have shown that there is a great deal that companies can learn about implementing QA and measurement from the relatively few companies who have programmes in place.

Chapter Five

An overview of real quality assurance practices in industry

1. Introduction

In this reconnaisance study I aimed to take a snapshot picture of the use and impact of QA and measurement in a variety of different companies. Indeed the objectives of this phase of the study were to: identify the real quality issues currently facing software companies; identify areas where future in-depth research could be most effectively targeted; perform some preliminary testing of my research hypotheses; further test the study methodology used throughout the rest of the research. The work presented in this chapter are the results from a survey I conducted at an IEEE Software Metrics Symposium [IEEE 1994].

In this preliminary study I looked at the QA and measurement programmes of sixteen different companies*. My aims were to:

- identify QA and measurement penetration within companies;
- analyse the impact of that penetration;
- compare penetration levels to certification;
- explore QA and measurement effects on software quality.

The chapter is organised as follows. In Section Two I discuss the penetration of quality controls in the companies, and I examine the impact on quality of those controls. Section Three explores the way in which various organisational factors affect quality. In Section Four I discuss quality certification and how it impacts on quality and on QA and measurement penetration. Section Five discusses the companies in the context of the Capability Maturity Model. In Section Six I summarise the main findings and conclusions of this study.

2. The penetration of quality controls

The anchor of many state-of-the-art programmes are basic quality controls such as standards, reviews and inspections. One of my aims with this phase of the study was to establish a baseline of quality control penetration. This section presents penetration level findings, while Section Four discusses these

* The small sample size means that I present the results of this study in raw numbers and do not use any statistical techniques on those numbers.

penetration levels in the context of quality certification. However, it is worth noting that the levels of quality control that I found are probably unrepresentative of the rest of industry, as by attending the symposium the companies already had an interest in quality

2.1 The use of standards

Table 5.1 shows that I found a relatively high penetration of standards in the surveyed companies. However, Table 5.1 also suggests that the use of standards is lower in testing than in areas such as documentation. [Vernier *et al* 1996] found a similar emphasis on documentation standards in the Hong Kong software industry as I did in the UK. It may be the case that some aspects of software development are easier to apply standards to than others, and that testing is relatively difficult to apply standards to and so is less standardised. Alternatively, it may be the case that standards are less useful in some areas than others. However, it may simply be a misunderstanding of the term documentation. Further research is necessary to establish the reasons for this imbalance in standards application.

Documentation standards			Code standards			Testing standards		
Never uses	Occasionally uses	Frequently uses	Never uses	Occasionally uses	Frequently uses	Never uses	Occasionally uses	Frequently uses
1	2	13	1	4	11	2	7	6

Table 5.1: Standards use in software development

2.2 The use of reviews and inspections

Table 5.2 shows that I found good review and inspection penetration in the companies. It is clear from my results, however, that even in these better than average companies, the use of reviews and inspections is lower than the use of standards. It is, of course, possible that standards must be used more often than reviews, as practitioners may use standards every day, but only participate in reviews every month. However, it might also be the case, even given the strong evidence in support of using reviews, that reviews are perceived as difficult and time consuming and so are not conducted as often as is optimal. More research is necessary to establish the reasons for this.

Specification review or inspection			Desgin review or inspection			Code review or inspection			Documentation review or inspection		
Never uses	Occasionally uses	Frequently uses	Never uses	Occasionally uses	Frequently uses	Never uses	Occasionally uses	Frequently uses	Never u ses	Occasionally uses	Frequently uses
1	6	9	1	7	8	2	8	6	2	4	10

Table 5.2: Review and inspection use in software development

2.3 The effects of quality control on quality

Table 5.3 shows that although I found basic quality controls in place within the surveyed companies, respondents did not believe that quality levels were consistently high. Indeed a quarter of respondents thought that high quality software was only produced *sometimes*. There are many possible interpretations of this finding and indeed there may be a discrepancy between perceived and actual quality levels.

However, my results suggest that while quality controls have an impact on quality they are not the only influence. Indeed I found that there can be high penetration of standards and inspections, but fairly low perceptions of software quality. Again further research is necessary to explore the impact of other factors on perceptions of quality.

How often high quality work is produced				
Always	Usually	Sometimes	Rarely	Never
0	11	5	0	0

Table 5.3: Software quality perceptions

3. The company culture

Basic quality control procedures clearly affect product quality. However, evidence from the literature suggests that broader organisational factors also affect quality. Indeed the organisational structure and ethos within which quality controls exist must influence the effectiveness of those quality controls. To explore this idea I asked respondents to comment on their companies' general approach and attitude to software development and to software quality. Table 5.4 presents my findings.

Company's approach to software quality				Company's approach to software development*			
very committed	No resources committed	lip service only	not Interested	highly structured	highly managed	highly creative	quality focused
5	10	1	0	9	6	7	10

Table 5.4: Perceived company attitudes

I suspect that practitioners' perceptions of their company's attitude towards quality is likely to affect the seriousness of their own approach to quality. In testing this suspicion, Table 5.4 shows practitioners

* Respondents could select a variety of answers, hence the total does not add up to 16.

believe that, although committed to quality, their company does not commit the resources necessary to improve quality. Indeed I found that only five respondents thought that their company was committed to quality and that it also provided adequate resources for quality improvement. The respondents who were most positive about their company's commitment to quality also claimed the highest level of quality control use. They also had slightly higher perceptions of their companies' software quality. Respondents in these positive companies claimed 25% more standards use and 13% more review use (with small samples the error rates are high).

As well as examining perceptions of company approaches to quality, I also asked respondents about general company approaches to software development. Table 5.4 shows that although respondents thought their company had several parellel approaches to software development, a quality focus was seen as most important during development. Creativity came well down the list.

4. The effect of quality standards on quality

Although there is a great deal of enthusiasm for companies to become quality certified, there has been little work done (with the exception of [Bazzana *et al* 1995]) in evaluating the penetration and impact of quality standards. In this study I begin to address this deficit. In this section I look at the kind of companies that seem to be implementing quality certification (I asked about any kind of quality certification) and I look at the effect that this has on quality control penetration and on perceptions of quality.

4.1 Companies and quality certification

		Certified	Not Certified
Size of company (people employed)	<100	0	1
	100 - 500	1	1
	500 - 1000	1	0
	1000+	7	4
Software engineering size (people employed)	10 - 100	0	2
	100 - 500	3	1
	500+	6	3
Applications developed	In-house	4	4
	Software House	2	2
	Public Sector	2	0
	Data Processing	1	2
	Safety Critical	2	2
	Research & Development	2	5

Table 5.5: Company features

Nine of the companies in the study had some sort of quality certification. Table 5.5 gives an overview of the characteristics of companies both with and without quality certification. It suggests that larger companies are more likely to have certification. It also suggests that the bigger the software engineering function the more likely a company is to have software quality certification (though this is probably related to the company's size too). This is supported by the findings of other studies [Vernier *et al* 1996]. I also found a curious pattern of application area certification. In particular a large proportion of in-house developments are not certified. This leads me to ask many questions about why companies do not subject their internal software to quality certification. It is also interesting that very little safety critical software seems to be certified. Major questions emerge from these results about the efficacy of quality certification.

4.2 Approaches to certification

Table 5.6 shows some very interesting results regarding the wider impact of quality certification. It is particularly interesting that there does not appear to be any more commitment to quality in certified companies than in non-certified companies. In fact the reverse could be the case. Two respondents from non-certified companies said that their company was *very committed* to quality compared to two certified companies. Furthermore the only respondent who said that their company *paid lip-service* to quality was from a certified company (though on such a small sample significance levels are poor).

Table 5.6 also shows that despite certified companies appearing to have no more commitment to quality than uncertified companies, they do tackle quality issues differently. For example, six certified companies were said to adopt *a highly structured* approach to quality compared to two non-certified companies. Generally there seemed to be a much greater emphasis on having a quality-focus, and adopting a creative approach to quality, in non-certified companies. There did, however, appear to be a perception that high quality software was delivered in companies with quality certification. However, morale was perceived to be higher in non-certificated companies, with four non-certified companies being said to have high morale compared to three certified companies. Of course these differences in results may just be chance or may be because of the different types of companies which tended to be certified in this sample, i.e. larger companies. Also with such a relatively small sample the error rates on these results are high. Nevertheless the results generate some interesting potential areas of future research.

		Certified	Not Certified
Company's attitude to quality	Very committed	2	2
	Lacks resources	6	3
	Lip service	1	0
	Not interested	0	0
	Other	0	1
Company's approach to quality	Highly structured	6	2
	Highly managed	3	2
	Highly creative	2	4
	Quality focused	5	4
How often a department produces high quality software	Always	0	0
	Usually	8	2
	Sometimes	1	2
	Rarely	0	0
	Never	0	0
	No response	0	2
Company's morale	Very high	1	1
	High	2	3
	Neutral	5	2
	Low	0	0
	Very low	1	0

Table 5.6: Quality culture

4.3 Quality controls and certification

Table 5.7 analyses quality control use compared to certification. It shows that there seems to be very little difference in standards penetration - although certified companies do seem to be using standards slightly more frequently. This is especially the case with documentation standards where eight certified companies said that they used documentation standards frequently compared to four non-certified companies.

Table 5.7 suggests a slightly higher penetration of reviews and inspections in certified companies. This slightly higher penetration may explain the higher quality perceptions that I have already mentioned (as the more inspections that are carried out, the more likely people are to be knowledgeable about quality levels, and the more likely software is to be of high quality). In particular certified companies seem to use code inspections more frequently than non-certified companies (four compared to one). However, I found no more certified companies with code inspections actually in place, the difference was only in the intensity of use. More research is required to establish the reason for this anomaly. One explanation could be that the quality systems in a certified company may be more transparent and so the respondents in these companies are more aware of the frequency of inspections, though, of course, it may be the case that there really are more code inspections taking place in certified companies.

	Certified	Not Certified
Frequently use standards:		
Documentation	8	4
Code	6	3
Test	4	2
Frequently use reviews/inspections:		
Specification	5	3
Design	5	2
Code	4	1
Documentation	6	3

Table 5.7: Quality control use

Some interesting results also emerged about the collection of measurement data. I found that certified companies were more likely to collect measurement data, three certified companies compared to one non-certified company. However, certified companies also said that a lot of the measurement data collected was not used. This is an area I decided to follow-up in later phases of the study.

Overall I felt that certified companies might be 'going through the motions', and that when it came down to quality mechanisms there was not very much difference in what companies actually implemented. The differences I observed were probably as a result of normal process maturity differences, rather than whether or not the company had quality certification. However, it did seem that part of that maturity process was achieving quality certification.

5. Capability maturity

I asked respondents to estimate their company's CMM level and Table 5.8 shows their ratings. My results suggest a high level of over-estimation. Given the fairly chaotic state of basic quality control that I observed in the companies, it is unlikely that only five of the companies are at level one of the model. This erroneous estimating could, of course, have many causes and a proper CMM assessment would be required to establish the true accuracy of these estimates. Nevertheless this serves as a good illustration of how over estimates about what is being achieved in terms of quality tend to occur.

Maturity Level					
One	Two	Three	Four	Five	Don't know
5	5	2	1	1	2

Table 5.8: Claimed capability maturity levels

6. Conclusions

The main aim of this phase of the study was to take a snapshot picture of the state of QA and measurement in industry, so as to identify further research areas, and test out the study methods. The study methods were successfully tested and further refined, and promising areas of further research were identified for follow-up. These include further exploration of:

1. The impact of certification.
2. The measurement process.
3. The factors which affect software quality.

The small scale of this phase of the study meant that statistically insignificant results were generated. Consequently, I was unable to test any study hypotheses during this phase of the study. Instead I used the results to refine my overall research hypotheses for follow-up in subsequent phases of the research.
The actual results from this reconnaissance study suggest that industry has an *ad hoc* approach to quality. Very few of the companies I surveyed demonstrated rigour and control in the way that quality was addressed. This was despite the fact that the companies which participated in this phase of the study are likely to be skewed in favour of an emphasis on quality. I infer from this that the rest of industry is more chaotic still.

In comparing companies with formal quality certification to those without it, I found little difference in the way that quality was tackled. Software quality was perceived to be not much higher than in companies without certification and many of the practices in the companies with certification seemed no better than in non-certificated companies. In particular, the use of basic quality controls (for example, the use of standards) appeared no more frequent in certificated companies. In addition, respondents from certificated companies were more likely to feel that less than half of the software measurement data collected by companies was used.

My results also suggest that a company's quality ethos influences how effective quality programmes are perceived to be. In particular there were lower software quality perceptions in companies which were seen to only be 'paying lip service' to quality. Significantly I observed this quality ethos in a company which had quality certification. The ethos in the majority of the companies I surveyed was said to be one where, although a commitment to quality existed, insufficient resources were made available to follow that commitment through effectively.

Overall the survey presents a snapshot of companies at a variety of quality maturity levels. This snapshot suggests that there is a lifecycle of quality maturity that is related to a number of factors (for example, size of company, quality ethos etc.). I follow-up the impact of some of these factors in studies reported in subsequent chapters.

Chapter Six

Practitioners, quality assurance and measurement: an industry-wide perspective

1. Introduction

In this chapter I present the first set of results from my main study. I look in detail at some aspects of QA and measurement identified as potentially promising in the previous reconnaissance phase of the research (discussed in Chapter Five). I follow up these promising areas by analysing the experiences of practitioners in four companies. I also test a sub-set of my initial study hypotheses and look in more detail at them within individual companies in the following three chapters. I concentrate particularly on the hypotheses about practitioners. In the conclusions of this Chapter I present a detailed analysis of how my results here produce evidence relating to these hypotheses. The data analysis discussed in this chapter has been applied to a relatively large sample of practitioners and so the results I present have very good statistical significance.

In this Chapter I also quantify the use and effectiveness of particular QA and measurement practices in the companies, and comment on the gap between what I would expect mature companies to be doing in terms of QA and measurement and what they were actually doing. I also discuss the views practitioners have about software quality generally and the use of QA and measurement specifically. I examine how the introduction and use of specific QA practices have been handled in the companies and I present the factors that I and practitioners identify as critical to a successful QA and measurement programme. I identify the QA and measurement factors that were relevant across all four companies. I follow this up by discussing company-specific factors in the following two chapters. None of the companies that participated in this phase of the study had software quality certification*. This meant I was unable to follow up the impact of certification at this stage. However, I do follow up in a later stage of the study, the results of which are discussed in Chapter Nine.

The chapter is structured as follows. Section Two discusses the general attitudes that practitioners have towards quality. Section Three discusses the penetration of quality control across the companies in the study. Section Four addresses the impact, on practitioners and on quality, of using those quality controls. Section Five looks at the broader organisational issues surrounding QA and measurement, in particular I look for evidence of good management practice within the companies. In Section Six I explore how different groups of practitioners think and react differently to QA and measurement, this discussion covers management seniority, gender, technical expertise and individual experience. I summarise and conclude in Section Seven.

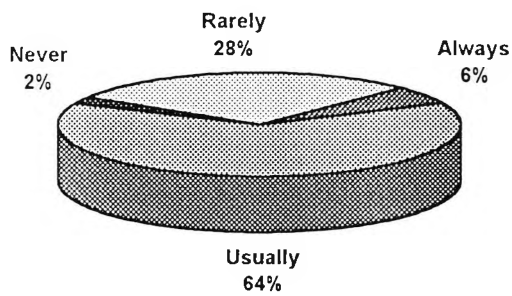
* One company did comply to a military standard.

2. Practitioner attitudes

Across all the companies in the study I found practitioners were keen to see software quality improve and seemed interested in the tools and procedures that could help achieve improvement. Indeed, overall, practitioners seemed very positive about the use of most quality-related mechanisms. There was also a broad cross-company consensus on how developing software within a quality-orientated framework was:

- effective (71% of the 178 respondents);
- efficient (67% of the 178 respondents);
- flexible (76% of the 178 respondents).

In addition, many practitioners were enthusiastic to see the use of more standards and procedures within their companies. Practitioners were also very positive about the benefits of software measurement, with 70% saying they thought the collection of measurement data was useful.



Do you produce high quality work?

Figure 6.1: Work quality

Figure 6.1 shows that, on the whole, practitioners were realistic about the quality levels they were achieving, with most claiming to develop high quality software *usually*. However, managers were over-represented in those claiming to *always* produce high quality work. This is interesting as [Jones 1994] suggests that technical staff perform more effectively than managers. In this study managers believed they were performing best. Indeed, throughout this study I found managers were more bullish than developers about all aspects of quality.

3. Quality control practices

My results show that a range of quality controls were regularly used in the four companies. These included inspections, reviews, standards, procedures, manuals etc. Furthermore, only 13% of practitioners said that they personally spent no time on quality-related activities. However, I confirmed my reconnaissance finding that standards are more widespread than inspections or reviews (10% of practitioners said they *never* used any standards compared to 23% who said they were *never* involved in reviews or inspections).

On the whole, the right people were using the right QA mechanisms. For example, 99% of programmers said that they used coding standards, while 74% of them were also involved in code reviews or inspections. I also confirmed that the most commonly used QA mechanisms were documentation standards. Almost everyone said they used documentation standards. Tables 6.2 and 6.3 show in more detail how much quality control practitioners said was taking place in a variety of lifecycle activities within the companies. However, I do not know how well these QA activities were being applied. Other studies show that in companies with apparently good quality control, actual practice is poor [Hatton 1995], [Neil 1994].

Standards	**Involvement level of respondent**					
(Frequency with which	Coding			Testing		
they are used)	Never %	Occasionally %	Frequently %	Never %	Occasionally %	Frequently %
Never	90	29	1	97	22	13
Occasionally	8	49	34	3	54	40
Frequently	2	23	65	0	24	47

Table 6.2: Penetration of standards

Reviews and Inspections	**Involvement of practitioner in development activities**								
(Frequency with	Specification			Design			Coding		
which they take place)	Never %	Occasionally %	Frequently %	Never %	Occasionally %	Frequently %	Never %	Occasionally %	Frequently %
Never	44	28	14	58	23	36	92	57	26
Occasionally	49	56	66	27	64	41	4	29	47
Frequently	6	15	20	13	13	24	4	11	27

Table 6.3: Penetration of reviews and inspections

The penetration of reviews and inspections varied between lifecycle phases, with less emphasis than I expected on code inspections. [Vernier *et al* 1996] also found this. There was, however, a range of other QA activities occurring. For example, 95% of practitioners said that they used a Quality Manual, 90% that configuration management was in place and 62% that measurement data was collected.

However, in the case of measurement, although practitioners had high perceptions of the amount and intensity of measurement they believed their company had implemented, in no case was the company's actual measurement process as far reaching or comprehensive as practitioners believed. While many practitioners said that their company collected a particular measure, I often could find very few or no people at all who were directly involved in the collection of some of these measures. For example, although 55% of practitioners said that function point data was collected, only 8% of practitioners said that they were involved in that measurement. If function points were being collected and used in any real sense, I would expect to find a higher participation rate than this. Similarly, although 30% of practitioners said that their company had a measurement database, none of the companies in the study officially had such a database. These anomalies were probably a result of the poor communication that I found within all the companies throughout the study.

4. The impact of quality control

My results suggest that the regular use of quality controls has a positive effect on practitioners' attitudes to quality. The regular use of standards, reviews and inspections made practitioners believe they were developing high quality software. Tables 6.4 and 6.5 show that this was particularly true where practitioners' work was regularly reviewed or inspected. These results contribute further evidence (though of a different type) to existing work which suggests that although reviews and inspections measurably improve software quality, the case for using standards is more difficult to substantiate [Fenton *et al* 1994]. Tables 6.4 and 6.5 also suggest that practitioners find the use of reviews least useful when applied to code (it is the only relationship on these tables which generated an insignificant Chi-square result), but most useful when applied to specifications. This finding goes some way towards substantiating the anecdotal claims of many practitioners that code inspections are not particularly useful (though more investigation is necessary to establish that this is not just a self fulfilling prophecy, *i.e.* if practitioners do not believe inspections are useful they will not put the effort into making them useful). I did, however, find some evidence of practitioners benefiting from using coding standards.

Frequency that high quality work is produced	**Standards used on work**											
	Specification			Design			Code			Documentation		
	Never %	Occasionally %	Frequently %	Never %	Occasionally %	Frequently %	Never %	Occasionally %	Frequently %	Never %	Occasionally %	Frequently %
Always	0	5	11	5	14	5	5	3	15	7	8	8
Usually	82	82	75	77	74	84	79	78	79	70	77	82
Rarely	18	16	14	18	12	11	16	19	6	23	15	10
Never	0	0	0	0	0	0	0	0	0	0	0	0

Figure 6.4: The effect of standards on quality

Frequency that high quality work is produced	Reviews performed on work: Specification			Design			Code			Documentation		
	Never %	Occasionally %	Frequently %	Never %	Occasionally %	Frequently %	Never %	Occasionally %	Frequently %	Never %	Occasionally %	Frequently %
Always	8	4	23	8	4	18	9	4	10	8	4	14
Usually	74	85	59	72	90	61	75	87	76	74	82	78
Rarely	19	10	18	20	6	21	16	10	14	18	14	8
Never	0	0	0	0	0	0	0	0	0	0	0	0

Figure 6.5: The effect of reviews and inspections on quality

Although conventional wisdom suggests that developers generally do not like using standards and inspections, I found that developers regarded them as beneficial. It appears that if developers have a standard, and a mechanism to check that work has met a standard, they feel confident that the quality of their work is high. Clearly for practitioners to do high quality work, and feel positive about the quality of that work, is good for morale and beneficial to companies. Practitioners appear to want explicit quality goals (i.e. standards) and formal affirmation of having met those goals (i.e. inspections). Without this formalism people seem less confident about how good their work is.

My results are an indirect measure of the impact of standards, reviews and inspections on software quality. Direct measurement of the quality of the software actually produced using these mechanisms would have to be carried out for a more conventional relationship between using quality controls and producing high quality software to be formally established. However, my results do support results from more conventional studies, and so I am led to believe that this indirect method of assessing quality has merit.

5. Organisational and implementational factors

The broader setting in which a QA and measurement programme is based will influence the success of that programme. Consequently I looked at a variety of organisational and implementation factors. In particular, I looked for evidence of good practice in the implementation of programmes.

5.1 Goals

Before any quality improvement can be achieved, companies must have clear improvement goals on which to focus. Contemporary quality models are predominately based on this premise (for example, GQM). Furthermore, practitioners must have a clear understanding of those goals and should, ideally, have played an active part in setting them. If a company is to attain its goals managers and developers must both also be committed to those goals. Other studies show the negative results of mismatching company and practitioner goals [Warner *et al* 1995]. I tested how effectively companies set goals and found major inadequacies.

Although 40% of practitioners said that their company's goals were not made clear to them, this was exemplified when I asked practitioners to rank five software development goals according to what they believed their company's goals were. Table 6.6 shows how confused practitioners are about their company's goals. It shows that there is low variance in the scores for each goal, which means that there was no agreement amongst practitioners about their company's software goals. Although it could be argued that this opaqueness could be a function of amalgamating all the companies' results, in the next chapter I show that this is not the reason.

Goal	Median ranking
Speed	1=
User satisfaction	1=
Low costs	1=
Reliability	1=
Conformance to requirements	1=

Table 6.6: Perceived company goals

Although practitioners seemed very confused about what their company was aiming for, Table 6.7 shows that practitioners were much clearer about their own goals. There is also a high level of agreement amongst practitioners about their personal goals. This result is supported by findings from Hong Kong [Vernier *et al* 1996], where practitioners were found to have very similar views on quality attributes.

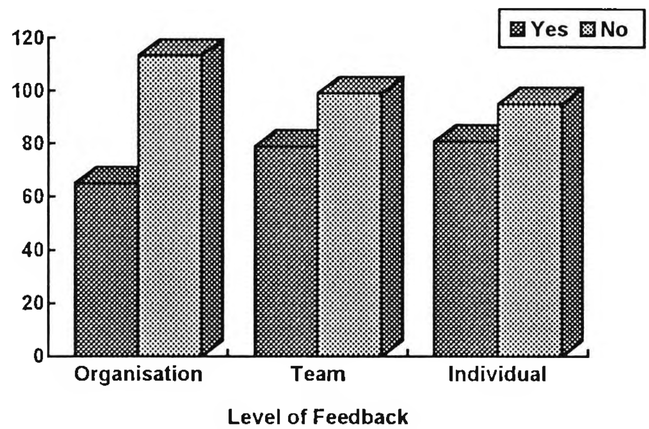
Goal	Median ranking
User satisfaction	1
Reliability	2
Conformance to requirements	3
Speed	4
Low costs	5

Table 6.7: Practitioner's own goals

Further analysis of this data shows that, although there was no particular difference in the personal goals of developers and managers, there was a difference in the goals that developers and managers think that their company has. For example, while managers think that their company rates reliability as a lower priority than developers do, there is no difference in how managers and developers personally rate reliability, and both rate it more highly than they believe their company does. This suggests that managers may find themselves with dilemmas. Managers tend to be more involved in setting goals: are they setting goals for the company that they are not personally committed to? Conflicting company and practitioner goals will make it difficult for companies to achieve their quality goals.

5.2 Communication

My results reveal that the companies in the study failed to communicate with practitioners about every aspect of quality. In particular Figure 6.8 shows how practitioners felt feedback on quality was missing at all levels. Other studies describe very negative outcomes of poor communication [Warner *et al* 1995].



Do you receive enough quality feedback?
(raw numbers)

Figure 6.8: Feedback

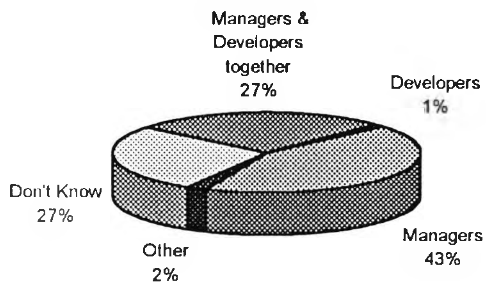
Lack of information emerged as a problem with the use of standards, reviews and inspections. When asked whether the data collected from code inspections was used, although 29% of practitioners said that it was, 58% did not know whether it was or not. This means that about a third of the people who said they were personally involved in code inspections and reviews did not know what happened to the data they generated.

This can either mean that the data generated during inspections is not actually used, and therefore the improvement process is weak, or that the process by which the data is used is not transparent to those involved and that communication is ineffective. Whatever the reason, effective quality improvement cannot be achieved without practitioners having a clearer understanding of the quality improvement process than was demonstrated.

The study revealed more poor communication between companies and practitioners regarding measurement. This problem was particularly acute with respect to communicating how measurement data was being used once it had been collected. Indeed 39% of practitioners said that no feedback at all was given once measurement data had been collected. Lack of communication has also been highlighted in other studies [Vernier *et al* 1996].

5.3 Participation

I found a very low level of developer participation in the design and management of quality improvement programmes. Figure 6.9 shows that only 28% of practitioners thought developers had any input at all to selecting quality measures. Furthermore, only 13% of developers but 41% of managers thought that developers had input.



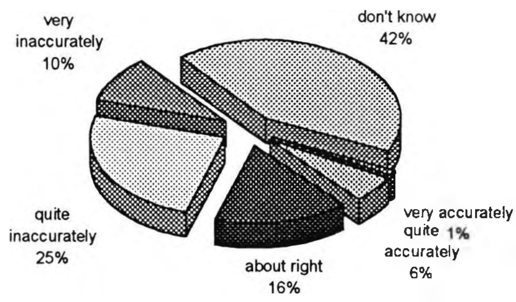
Who decides on what software measurement data will be collected?

Figure 6.9: Participation

I found that direct participation in all manner of QA mechanisms seems to improve practitioners' attitudes to those mechanisms. Just as practitioners felt more positive the more they used reviews and inspections, the more time practitioners spent working with measurement, the more positive they felt about measurement. For example, only 17% of the practitioners who spent less than one hour per week on measurement felt that measurement was a *very useful* activity, compared to 40% of practitioners who spent more than one hour per week on measurement (however with a Chi-square result of 0.38, such a weak statistical relationship needs more research).

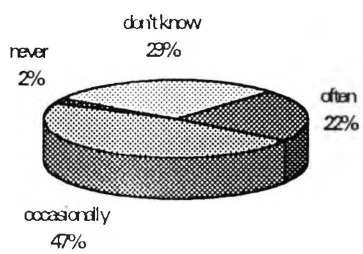
5.4 Integrity

Practitioners were also not confident about the integrity of some aspects of their quality system. Indeed Figures 6.10 and 6.11 show that a significant number of practitioners were unhappy with the accuracy and integrity of measurement data. Furthermore, Figure 6.12 shows that many practitioners also felt that the data collected was not the right data to collect.



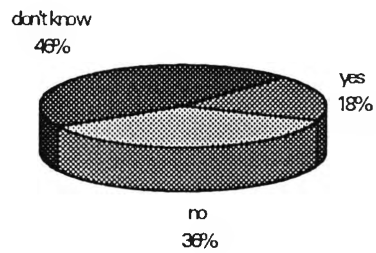
How accurately is measurement data recorded?

Figure 6.10: Data accuracy



Do you think that there is any manipulation of software measurement data in your company?

Figure 6.11: Data manipulation



Do you consider that your company collects the right software measurement data?

Figure 6.12: Appropriate data

I believe that the way in which QA mechanisms are implemented determines how positively practitioners perceive them and, therefore, how successful they will be. Thus, the way data is collected and used appears to play an important role in determining practitioners' attitudes to measurement. Table 6.13 supports my feelings by showing that practitioners who got feedback on measurement were more likely to be positive about measurement.

Are metrics useful?	**Is metrics feedback provided?**		
	Yes	No	Don't Know
Yes (%)	28	34	38
No (%)	6	50	44
Don't Know (%)	22	56	22

Table 6.13: The perceived usefulness of measurement

Overall practitioners were negative about software measurement when they believed that:

- measurement data was collected inaccurately;
- no feedback was given on the data collected;
- managers had sole input to developing measurement;
- the wrong measures were collected;
- the data was manipulated;
- the data was used secretly, for individual assessment or to affect salary.

However, I also found that what practitioners believed was happening to measurement data was not always accurate. For example, although no company admitted to using data to determine salaries, eight practitioners thought data was being used for that purpose. The fact that some practitioners *thought* that measurement data was being used for dubious management practices is likely to have a negative impact on their perceptions of measurement. Such negative perceptions are likely to have a detrimental impact on a company's measurement activity. Indeed, how practitioners *think* data is being used is as important as how data is actually being used. This, again, highlights how important it is to do the right things and communicate to practitioners that the right things are being done.

5.5 Morale

My results suggest that there is a relationship between high morale and what practitioners felt about QA and measurement. For example, practitioners who were positive about QA and measurement tended to have higher morale[1]. I observed a similar relationship between job satisfaction and attitudes to QA and measurement.

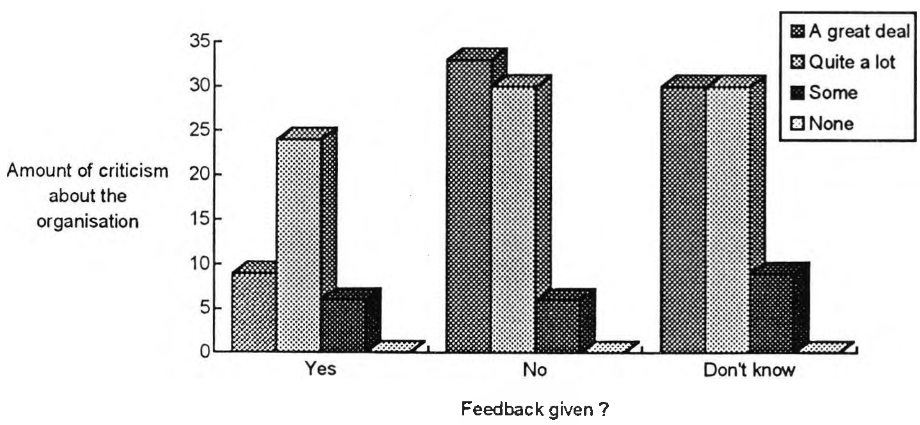


Figure 6.14: The relationship between measurement feedback and general morale

[1] Practitioners were asked *"To your knowledge, how much criticism, by employees, is there about your company?"*. This is a standard question that provides a general index to morale.

I also observed a relationship between practitioner morale and the way in which QA and measurement had been *implemented*. For example, 13% of practitioners who said that feedback was given on measurement data had low morale compared to 46% who said that no feedback was given (see Figure 6.14). A similar relationship existed between low morale and practitioners feeling that they were not directly involved in the initial setting up of measurement, or that measurement data was being *manipulated*. However, the Chi-square values of these results were not significant enough to enable the null hypothesis to be rejected at this stage.

One interpretation of these results is that poorly implemented QA and measurement not only negatively affects attitudes to QA and measurement, but also has a wider negative affect on practitioner morale. Another interpretation is that practitioners with existing poor morale will also be negative about QA and measurement. More research is necessary to identify whether negative attitudes to QA and measurement is the cause of poor morale or whether it is simply an effect of it.

6. Practitioner issues

I explored whether different groups of practitioners feel differently about quality and about the use of QA and measurement.

6.1 Managers and developers

Throughout this research I found that among all staff, managers were most positive about all aspects of quality. They were most positive that they were doing high quality work and they were most positive about the value of QA and measurement. This developer/manager gap is particularly well illustrated by attitudes to measurement. Table 6.15 shows that managers were most positive about measurement. Managers were also much less likely than developers to believe there were any problems with data accuracy or measurement design. For example, 34% of managers said that data was accurate, compared to 10% of developers. Similarly 31% of managers thought that feedback was given on data, compared to only 15% of developers (Table 6.16). Managers also seemed confident that data were used constructively and responsibly whereas developers were more likely to say that data were used for staff appraisal and to influence salary.

Furthermore, the more senior the manager the keener on measurement they usually were. Table 6.16 also shows that the most senior managers seemed to have the most positive view of how measurement was working, while middle managers (the people most directly involved in measurement) were less positive.

The Chi-square values generated by the analysis of manager and developer responses were very significant. Consequently the null hypothesis for the above relationships can be confidently rejected.

How useful an activity do you consider the collection of software metrics data to be?	Senior Managers %	Quality Managers %	Middle Managers %	Developers %	Other %
Very Useful	44	22	13	9	36
Quite Useful	46	78	47	51	41
Not Useful	8	0	30	26	9
Not at all Useful	0	0	7	10	0
Don't Know	3	0	3	4	14

Figure 6.15: Attitudes to measurement

Is feedback on metrics provided?	Senior Managers %	Quality Managers %	Middle Managers %	Developers %	Other %
Yes	38	22	30	15	9
No	40	67	50	32	36
Don't know	22	11	20	54	55

Figure 6.16: Feedback on data

Tables 6.17 and 6.18 show that similar proportions of management and development staff thought that measurement data was manipulated, and that managers did the manipulation. Other studies have also found deliberate mis-recording of data [Kitchenham & McDermid 1986], and practitioners who wanted to retrospectively alter projections and estimates [Lytz 1995]. Indeed [Kitchenham 1993] believes that practitioners are very likely to lie via measurement data. [Glass 1994] says that there is an incentive for this, as senior managers often do not want to hear the truth or will not listen to it. [Hetzel 1993] believes that this problem is much more likely to occur in a top-down as opposed to a bottom-up company hierarchy.

Is there any manipulation of metrics data?

	Often	Occasionally	Never	Don't know	Total
Managers	22	49	2	27	100%
Developers	23	44	3	31	100%

Table 6.17: Level of data manipulation

Who do you think manipulates measurement data?

	Developers	Managers	Neither	Both	Don't know	Total
Managers	14	37	3	24	23	100%
Developers	9	36	2	21	31	100%

Table 6.18: Source of data manipulation

There were, however, some disturbing answers from some managers. For example, 22% said that they thought measurement data was being used *secretly*; 6% said it was being used for assessing individuals; 5% said it was being used to affect salary. If managers were saying that they thought these practices were taking place, they probably were and there is plenty of anecdotal evidence to support this. Furthermore, as Table 6.19 shows managers were also much more likely than developers to be intimately involved in data collection, so they are likely to know how data is being used.

	Hours per week			
	< 1*	1 - 5	6 - 10	> 10
Managers (%)	67	27	4	2
Developers (%)	90	9	1	0

Table 6.19: Time spent on data collection

6.2 Measurement and quality assurance staff

A total of nine practitioners classified themselves as *Quality Assurers*, *Controllers* or *Managers*. Although this group were all positive about the use of QA, they did have some worrying things to say about measurement. This included, of the nine staff in the study:

- only one saying data was collected accurately;
- only two saying feedback was given on data;
- one saying they did not know who initially designed the measures in use;
- six saying they thought the wrong data was collected;
- eight saying they thought data was manipulated;
- one saying that data was being used to assess individuals (none thought that it was used to affect salaries).

Clearly quality assurance staff think that measurement is being handled somewhat dubiously. Given that quality assurance staff tend to be highly involved in measurement, it is worrying that they seem unable to improve their company's measurement activity. Alternatively, it could be that some quality assurance staff think that it is legitimate and constructive to use data to assess individuals.

Overall the results obtained from this group of staff were perplexing, and generated more questions than they answered. However, overall quality assurance staff appeared both optimistic about the long term, but aware of current weaknesses.

* Unfortunately I did not ask whether no time was spent on data collection..

6.3 Age and experience

On the whole my results suggest that those practitioners with most experience are likely to feel positive about QA and measurement. In particular I found practitioners most positive about QA and measurement tend to be:

- well established within a company (usually having worked there for over ten years);
- older;
- male;
- well qualified (usually having at least a degree);
- a line manager;
- highly motivated and positive about the company.

This profile contrasts with the conventional wisdom describing those people most likely to accept new working methods and practices. However, this description of a 'QA and measurement friendly' practitioner may simply describe managers.

6.4 Gender

Approximately 10% of software developers are women. My research suggests that women have subtly different perspectives on software quality to those held by men[2]. This difference was evident in the following areas:

6.4.1 Perceptions of work quality

Women developers seemed less satisfied with the quality of the work that they produce: 27% of women said that they *rarely* produced good quality work, compared to just 12% of men. Men were twice as likely to say that they *always* produced good quality work. This result generates many questions. Do women developers have higher standards than men? Are women more objective and less bullish about the quality of their work?

6.4.2 The effectiveness of formal QA mechanisms

While women and men each spent equal amounts of time on quality-related activities, the study suggested that women were more critical of formal QA mechanisms. In particular they were less convinced about the effectiveness of standards and inspections, and about the effectiveness of software measurement (34% of women thought that measurement was not useful compared to 23% of men). These results may be related to the artificially low perceptions that women have about the quality of their work. Perhaps they are concerned that such mechanisms will reveal what they

[2] The relatively small number of women participating in this study means that most of this data analysis did not generate Chi-square values that would enable the null hypothesis to be rejected confidently.

perceive to be their low quality work? Maybe they are just uncomfortable that their work (and therefore themselves) will be 'in the spot light' during, say, an inspection?

It is generally believed that women have a more organised and methodical approach to tasks and, therefore, would favour using inspections etc. It seems possible, therefore, that the QA mechanisms which can be construed as assessment-based are unpopular with women. Maybe in the past such assessments have been used to victimise people?

6.4.3 Satisfaction with the way QA mechanisms are used

Overall women were more critical than men about how mechanisms for quality were being used in practice. For example, when asked what they thought about the use of measurement data, more women than men thought that data is:

- not accurate (14% of women thought the data collected was accurate, compared to 25% of men);
- manipulated (78% of women compared to 67% of men);
- not used constructively (28% of women compared to 18% of men).

Although women did not have as much faith in the way that QA mechanisms were being used as men, women were generally more satisfied with their job, and with the company that they worked for. So their perceptions did not seem to affect their morale.

6.4.4 Feedback on quality

When developers were asked whether they received enough feedback on quality some interesting trends emerged. Men and women agreed that there was not enough overall feedback on their company. However, women were more satisfied with the level of team-based feedback, and a lot more satisfied with the level of individual feedback (61% of women said that they were satisfied with individual feedback compared to just 43% of men). In the context of women's poor work quality perceptions some interesting questions emerge. Do women developers need less feedback than men? Or are women more satisfied because they are actually getting more feedback anyway?

6.4.5 Knowledge about quality

All of the study companies had big problems communicating with staff about quality. The majority of all staff in the survey (52%) did not know whether their work was externally quality certified and 10% of all staff did not know whether or not they had a Quality Manager. In general, women developers were much more likely to answer *don't know* to a question about QA mechanisms than men (this was not the case for every question). In some cases women were either slightly more likely to know whether a particular mechanism was in place or not, and therefore answer the question correctly, or to answer *don't know*. This suggests that if women did not know whether, say, their company had a quality manual they were unlikely to guess. On the other hand, men were slightly less likely to answer *don't know* and then more likely to get the answer wrong. This means that women developers may appear less knowledgeable than their male counterparts about quality, but they are probably just being more honest about what they do know than men. In addition women are also more accurate in what they do say, as they seem less likely to guess.

7. Conclusions

7.1 General findings

The most important findings of this phase of the study are about communication and participation during the implementation of QA and measurement. I show that poor communication about QA and measurement, negatively affects the attitudes that practitioners have. This is particularly true of communication about how measurement data will <u>and will not</u> be used. Similarly, the amount of perceived and actual developer participation in QA and measurement also affects attitudes. The fewer developers that are seen to participate, the more negatively QA and measurement will be generally perceived by developers.

The study also identifies an important area of additional research examining further the relationship between QA and measurement implementation and company morale. Although I found the two seemed to be related, it is critical to identify whether it is that existing low morale causes negative perceptions about QA and measurement or that low morale is the result of poor implementation.

The study also revealed a worrying level of goal incongruence. I observed this between developers and managers, and between managers, developers and companies.

Overall I found that practitioners were generally optimistic about the use of QA and measurement and realistic about the current state of quality. However I also found that the way in which QA and measurement was being managed could be improved. Basic implementation errors were very obvious. For example, companies did not seem to be involving developers enough during the implementation process, nor did they seem to be providing enough feedback to developers. These errors would have been avoided if the advice in the existing QA and measurement literature been referred to by companies.

7.2 Evidence relating to the study hypotheses

In this chapter I looked for evidence to support the following practitioner hypotheses:

i. *Managers are enthusiastic about QA and measurement.*

I found evidence suggesting that management seniority was related to enthusiasm for QA and measurement, however not all managers were equally enthusiastic with middle mangers showing lower enthusiasm (see section 6.1).

ii. *Developers resist the implementation of QA and measurement.*

I found evidence suggesting that although developers did not seem to actively resist QA and measurement, developers were generally less enthusiastic about QA and measurement than managers (see section 6.1).

iii. *Younger practitioners who are more highly qualified will be keener on QA and measurement.*

I could find no evidence to support this hypothesis, indeed I found evidence suggesting that older people were keener (see section 6.3) but that keener people did tend to be well qualified.

iv. *Practitioners with particular technical skills will be more positive about QA and measurement than others. For example, analysts will be more positive than programmers.*

I found no evidence to support this hypothesis, there appeared to be no difference in attitudes between people with different technical skills.

I also tested the following company hypothesis:

i. QA and measurement is more likely to be successful where morale in a company is already quite high and practitioners generally feel positive about the company.

I found evidence of a relationship between company morale and practitioners feeling positive about QA and measurement and measurement success. However further work is needed to establish cause/effect, *i.e.* whether morale is the dependent or independent variable.

ii. QA and measurement that is implemented ineffectively will have widespread counterproductive repercussions throughout the company.

This hypothesis is related to the previous hypothesis and it too needs more evidence before a conclusion can be drawn.

Chapter Seven

Practitioners and quality assurance: company case studies

1. Introduction

In this chapter I explore in more depth some of the issues discussed in Chapter Six. In particular I examine QA in the context of individual company case studies. I go on to discuss measurement in the these same companies in the next chapter. Some characteristics of QA seem to be industry-wide, for example, the importance of transparency and effective communication channels. However some quality issues are directly related to individual companies and the way in which they have implemented QA. In this chapter I look at these company-specific issues. I also test a sub-set of the company and implementation hypotheses that I presented in Chapter One. I discuss how my findings in this chapter relate to those hypotheses in the conclusions section.

In this part of the study I use the two companies with the most mature and well established QA programmes as my case study companies. Full details of the two companies, and the study methods I used on them, are given in the Study Methods chapter (Chapter Two).

The chapter is organised as follows: Section Two examines the QA programmes in the two companies, concentrating particularly on: the quality infrastructure; the penetration of quality controls and goals for quality. Section Three analyses the company culture within which the two case study QA programmes are set. Section Four provides the main conclusions of this part of the study.

2. The quality programmes

The basic structure of both QA programmes seemed adequate (although neither was ISO certified). Both companies said that they had:

- lifecycle coverage of standards, reviews and procedures;
- tool support for software development and project management;
- configuration management;
- a measurement programme;

- a quality manual;
- a quality manager.

I also found evidence that many practitioners felt actively involved in QA, indeed Table 7.1 shows that this was particularly the case in Information Systems.

Time on QA	Percentage of practitioners	
	Embedded Systems	Information Systems
Frequently	25	30
Occasionally	50	56
Never	25	12

Table 7.1: The time practitioners said they spent on QA activities

In the rest of this section I focus on the penetration of quality controls such as standards, reviews and procedures in both companies. I also consider the issue of setting quality goals in each company.

2.1 Quality controls

2.1.1 Penetration

I found the use of quality controls was generally low, and was lower in practice than each company claimed. Even quality controls known to have significant cost benefit had low penetration.

Lifecycle phases	Work that is *never* reviewed				Work that is *frequently* reviewed			
	Embedded Systems		Information Systems		Embedded Systems		Information Systems	
	Activity (Percentage of Practitioners never involved in lifecycle activity)	Review (Percentage of Practitioners whose work is *never reviewed)*	Activity (Percentage of Practitioners *never involved in lifecycle activity)*	Review (Percentage of Practitioners whose work is *never reviewed)*	Activity (Percentage of Practitioners frequently involved in lifecycle activity)	Review (Percentage of Practitioners whose work is *frequently reviewed)*	Activity (Percentage of Practitioners frequently involved in lifecycle activity)	Review (Percentage of Practitioners whose work is *frequently reviewed)*
Specification	28	42	29	29	20	10	20	7
Design	5	32	25	38	41	20	18	12
Code	5	32	22	46	69	22	55	20

Table 7.2: Inspection and review penetration

Table 7.2 shows the penetration of reviews and inspections at three phases of the lifecycle in both companies. Practitioners were asked how much of their work was subjected to review or inspection. The

'Activity' column of Table 7.2 shows the involvement of practitioners at each phase in the lifecycle and the 'Review' column shows the penetration of reviews at those lifecycle phases. The difference between the two columns illustrates the review 'shortfall', *i.e.* although 28% of the practitioners were *never* involved in specification, 42% *never* had their specification work reviewed. I have construed from this that the difference between these figures (14%) gives a crude indication of the shortfall in work not getting reviewed. More evidence is shown in the columns detailing work *frequently* reviewed, where, for example, although 20% of the practitioners are frequently involved in specification only 10% of work is reviewed.

2.1.2 Quality control problems

I discovered the following about quality control penetration at the two companies:

- **Quality control shortfalls**
  A significant amount of work at every lifecycle phase is not inspected or reviewed. Furthermore, the review shortfall gets wider further up the lifecycle, so that more work is reviewed at specification than coding.

- **Ineffective deployment of quality controls**
  Despite empirical evidence demonstrating that reviews and inspections are a highly cost effective quality controls neither company used them widely. On the other hand standards were widely used in both companies throughout the lifecycle. For example, 86% of the practitioners at ES who spent most of their time coding, said that they *frequently* used coding standards, but only 14% *frequently* had their work reviewed. The picture was similar at IS. Furthermore 75% of ES practitioners whose main task was testing, *frequently* used test standards. 50% in IS said the same. The basis of either companies' preference for standards is not clear. I suspect that companies consider standards to be easier and cheaper to implement, even though this is not borne out by the empirical evidence.

- **Deviation from the 'official' position**
  I found a significant gap between the claimed and the actual use of quality controls. Senior managers at both companies said that standards were *always* used at all stages of the lifecycle. Senior managers at IS said that reviews were *always* performed on all lifecycle deliverables. Managers at ES said that reviews were *almost always* performed on all lifecycle deliverables. Table 7.2 shows that when I asked the practitioners themselves, the actual penetration of quality controls was far poorer than managers led me to believe. This deviation from 'the official' quality system is probably common; the SMARTIE project [Fenton *et al* 1994] discovered objective evidence of this. The underlying reasons for this deviation are, however, unclear.

2.1.3 The impact of quality controls

I also analysed the relationship between quality control penetration and perceptions of quality in both companies. Practitioners at both companies had relatively high quality perceptions of their work, whilst quite low quality control penetration.

The basis of these high software quality perceptions were not clear. Poor review and inspection penetration levels do not seem to adversely affect perceptions of quality. Although ES had comparatively low penetration, quality confidence levels in this company were very high. This would suggest that practitioners in this company received quality affirmation from other sources: their safety critical software did not fail disastrously and publicly. Alternatively, ES's relatively low inspection penetration may mean that practitioners are simply not aware of the quality of their work and so have an inflated view of its quality (though this is a more likely scenario if maintenance is done by separate departments).

On the other hand IS had quite good inspection penetration and correspondingly positive quality perceptions. This is to be expected. Practitioners' work is submitted for quality inspection, consequently those practitioners can be confident about the quality of their work.

2.2 Quality goals

I looked at how the two companies were performing in terms of quality goals. I analysed practitioners' own software quality goals and compared them to what practitioners thought the goals of their company were. Practitioners were asked to rank five quality goals according to what they thought their company's quality goals were and according to their own personal goals (1 = high priority; 5 = low priority); a median score for each goal was then calculated. As I now report, there was a disturbing level of goal confusion and variance in both companies.

2.2.1 Goals at Embedded Systems

	Company goals			Practitioners' goals		
Goal	Overall Ranking	Manager Ranking	Developer Ranking	Overall Ranking	Manager Ranking	Developer Ranking
Low costs	1	1	2	5	5	4
Conformance	2	2	4	2	2	3
Speed	3	3	3	4	4	5
Reliability	4	4=	1	1	1	1=
User satisfaction	5	4=	5	3	3	1=

Table 7.3: Embedded Systems' goals

Company goals at ES

Table 7.3 shows there was very little agreement amongst practitioners about company goals, although there is more agreement amongst managers than developers. This confusion about the company's software development goals is not surprising given that 60% of practitioners in this company said company goals were not made clear to them (Table 7.6).

Table 7.3 also shows that, overall, practitioners in ES think that low costs are the company's highest priority. This is a disturbing result from a high profile, safety critical equipment supplier. Furthermore, developers believe the company rates reliability more highly than managers do (ranked one by developers but equal fourth by managers). Whatever the explanation for this manager/developer disagreement it is worrying that managers themselves think that reliability should have the highest priority and yet perceive reliability to have a low company priority.

Managers in ES agree that the company ranks user satisfaction as least important. Although this potentially alarming result may be mitigated by the embedded nature of the application area, it must be noted that practitioners themselves rate user satisfaction as a very important goal.

Practitioners' goals at ES

Table 7.3 shows that there is a great deal more agreement on personal quality goals amongst practitioners. Nevertheless there are some notable differences in personal goals between managers and developers. For example, developers rate user satisfaction higher than managers; managers rate conformance to requirements more highly than developers; managers rate low costs as a less important goal than developers.

It is clear in ES there is a gap between what practitioners believe the company's quality goals are and what practitioners own quality goals are (they are almost inverse to each other). Such goal incongruence is likely to make quality improvement difficult.

2.2.2 Goals at Information Systems

Goal	Company goals			Practitioners' goals		
	Overall Ranking	Manager Ranking	Developer Ranking	Overall Ranking	Manager Ranking	Developer Ranking
User satisfaction	1	1	1	1	1	1
Speed	2	2	2	4	4	4
Reliability	3	3	3	2	2	2
Conformance	4	4	4	3	3	3
Low costs	5	5	5	5	5	5

Table 7.4: Information Systems' goals

Company goals in IS

There was a great deal more agreement amongst practitioners on what their company's software quality goals were. Indeed Table 7.4 shows that managers and developers ranked their perceived company goals identically. Overall, practitioners thought user satisfaction was a company priority. Practitioners also agreed low costs had a low company priority.

Practitioners' goals in IS

There was a high level of agreement here about practitioners' own quality goals. They had a very similar pattern to perceived company goals (agreement was evident between highest and lowest priorities). In addition there was almost complete agreement on personal goals between managers and developers.

3. The company culture

I believe that the efficacy of the company infrastructure has an impact on the effectiveness of a QA programme and, therefore, on the quality of software produced. Since practitioners are ideally placed to comment on the state of their company's culture and infrastructure, I asked them what their company's culture and infrastructure was like and how effective they thought they were.

How often do you think you work efficiently?	Percentage of practitioners	
	Embedded Systems	Information Systems
Always	0	1
Usually	60	69
Rarely	40	29
Never	0	1

Table 7.5: How often individuals think that they work efficiently

Overall I found that IS practitioners were happier with their company infrastructure than ES practitioners. Furthermore, morale at IS was significantly higher with 41% of practitioners selecting the lowest morale indicator compared to 60% of practitioners at ES. However, Table 7.5 shows that practitioners felt that the infrastructure at neither company allowed them to work efficiently, although practitioners at ES were more negative than practitioners at IS.

How often are company goals made clear to you?	Percentage of practitioners	
	Embedded Systems	Information Systems
Always	5	7
Usually	35	53
Rarely	50	36
Never	10	4

Table 7.6: How often company goals are made clear to individuals

Table 7.6 shows that both companies have a problem communicating to practitioners. Again, ES has the biggest problem. This is further evidence of what I saw in Tables 7.3 and 7.4.

Percentage of practitioners satisfied with various levels of feedback					
Embedded Systems			Information Systems		
Company	Team	Individual	Company	Team	Individual
18	15	12	25	50	57

Table 7.7: Satisfaction with levels of feedback

I also found that practitioners were very unhappy with feedback levels (this was more of an issue at ES). This is confirmed in Table 7.7 which considers feedback at three different levels.

4. Conclusions

4.1 General findings

Despite strong evidence that the systematic use of reviews and inspections is cost effective, the two companies discussed in this chapter (one of which produces safety critical software) were *not* incorporating them effectively into their development process. Moreover, the penetration of these basic quality controls was much lower than senior managers believed and claimed. It is not clear if this shortfall is a result of poor technology transfer (companies not knowing of the benefits), or companies not believing that there really are benefits. Either way the companies are deciding not to allocate effort to reviews or inspections while they are happier to allocate effort to implementing development standards. However, senior management in both companies were keen to *be seen* using reviews and inspections. This theory and reality gap could prove difficult for customers of software produced by suppliers not third party accredited. Such customers are likely to find it impossible to establish how much of the official QA programme is actually practised during production.

The results also show that each company had varying degrees of company and practitioner quality goal incongruence. This confusion about what the quality of the final software should be will make producing software at the right level of quality difficult. Again the particular goal incongruence difficulties that ES exhibited makes compromised safety critical software more likely.

Furthermore, I discovered that, contrary to popular belief, practitioners do not have a stereotyped 'wish list' for software development. In neither company did practitioners cite a clichéd list of quality goals, but rather seemed to tailor their own goals to their specific environment. This is important for managers. Practitioners are often excluded from influential goal setting initiatives because it is generally thought they have unrealistic goals. This study suggests that they do not.

The company goal analysis also suggests that companies want everything to be a priority. In neither company was there a clear consensus on company goals. Practitioners cited a variety of goals as their company's priority. This suggests that the companies were trying to emphasise too many goals, instead of being clear about real priorities.

The context in which a QA programme is set is important. A QA programme does not operate in isolation and so getting the company environment right, before even starting a QA programme, is the ideal. The companies in this study seemed to have some way to go in getting their company environments right. The results suggest that the companies are failing to provide a constructive working environment. Many basic tenets of management science are broken by both companies. In particular, basic communication within the two companies seems very poor. As a result of this many practitioners feel negative and de-motivated about their company. Mass staff de-motivation will make it difficult to optimise QA mechanisms. It seems that senior managers think that a successful QA programme can be instituted without considering the effectiveness of the company's overall framework. While practitioners in both of the companies believed that a number of deep company problems existed, senior management were primarily interested in developing the QA system rather than the overall company approach. Such a strategy is only likely to have limited success.

4.2 Evidence relating to the study hypotheses

4.2.1 Hypotheses about the company

i. *It is common for companies not to practice what they claim to in terms of QA and measurement.*

   I found strong evidence to support this hypothesis. In section 2.1 I show that quality control penetration is low in both companies. This is in direct contrast to what company mangers in each company told me at the beginning of the study. Managers in both companies claimed a very high review rate and also claimed that standards were regularly used. I found evidence suggesting this not to be the case.

4.2.2 Hypotheses about the implementation process

Given that I found no evidence that either company had an effective QA programme some of my hypotheses were impossible to test at this point in the study. These include:

i. *The goals of QA must be clear to practitioners if the programme is to succeed.*
ii. *Effective QA must be transparent to practitioners if the programme is to succeed.*
iii. *Feedback on QA and measurement needs to be provided if the programme is to be successful.*

However, I found no evidence that either company: made its goals clear to practitioners (see section 2.2); had a transparent process (see section 3) or provided feedback (see section 3). Although there is no firm evidence, the inference could be drawn that the lack of clear goals, the lack of transparency and the lack of feedback in each company contributed to the ineffectiveness of the two QA programmes.

Chapter Eight

Practitioners and measurement: company case studies

1. Introduction

My aim in this chapter is to present the critical success and failure factors associated with the measurement programmes in the two case study companies I examined. Although both of the case study programmes were structurally similar, one was judged to be successful (by the company and me), whereas the other was judged as not yet successful (by the company's own measurement group and by me). It is unusual that I was able to report on an unsuccessful as well as a successful measurement programme. With a few notable exceptions (such as [Jeffery & Berry 1993]), only successful programmes are generally reported. This is despite the fact that anecdotal evidence suggests that many measurement programmes are unsuccessful.

In this chapter I particularly:

1. Report the measurement implementation strategies each company adopted.
2. Quantify the extent to which various measures were implemented in both companies.
3. Describe the on-going management of the measurement programmes.

In this chapter I look at many of the initial study hypotheses (as presented in Chapter One). I address some hypotheses for the first time, and also uncover further evidence in support of some hypotheses already examined. I summarise how the data presented in this chapter relates to the study hypotheses in the conclusions section of this chapter.

Details of how I conducted this part of the study can be found in the Study Methods chapter (Chapter Two).

The chapter is structured as follows. Section Two discusses each company's measurement programme within the context of established success factors identified in the literature. Section Three analyses the case study measurement programmes in terms of other assessment criteria, for example, how the attitudes of practitioners to measurement affects the success of a measurement programme. Section Four describes how each measurement programme has evolved since the study. Section Five summarises and concludes.

2. Key success factors

Although the managers of the measurement programmes in both case study companies agreed that good measurement implementation was important, I discovered that many of the key recommendations made by experts (as discussed in Chapter Three) had been ignored. Table 8.1 summarises my findings. It shows which of the recommendations have been used by each company (✓ shows that the recomendation has been used and × shows that it has not been used).

Implementation Factors	ES	IS
Consensus Recommendations		
Incremental implementation	✓	×
Well planned measurement framework	✓	×
Use of existing measurement materials	✓	×
Involvement of developers during implementation	✓	×
A measurement process that is transparent to developers	✓	×
Useful measurement data collected	✓	×
Feedback to developers	✓	×
The data is seen to have integrity	✓	×
Measurement data is used and seen to be used	✓	×
Commitment from project managers secured	✓	×
Use automated data collection tools	✓	×
Constantly improving the measurement programme	✓	×
Internal measurement champions used to manage the programme	✓	×
Use of external measurement gurus	×	×
Provision of training for practitioners	×	×
Other Recommendations		
Implement measurement at a level local to the developers	✓	×
Implement a central measurement function	×	✓
Measurement responsibility devolved to the development teams	✓	×
Incremental determination of the measurement set	✓	×
Collecting data that is easy to collect	×	×

Table 8.1: Implementation factors

If the views of the experts are to be believed, then a cursory examination of Table 8.1 suggests that ES were more likely to have a successful programme than IS; they generally adhered to more of the proposed success factors than IS. Indeed ES's programme was the successful one. ES's programme was an integrated part of the development process, the data it generated was used in both day-to-day technical decisions and project management decisions. ES's programme was accepted by practitioners and managers alike and was about to be implemented throughout the company. On the other hand IS's programme was much less successful. Neither managers nor developers were particular impressed by its performance, with even the managers of the programme disappointed with it. It had made minimal impact on decision-making and was not really integrated into the development process. However, at least IS had made a start at measurement which is more than can be said for most companies. Indeed [Pfleeger 1993] argues that any measurement is better than no measurement and that initiating a programme is often the most difficult obstacle to overcome.

The rest of this chapter is devoted to a more detailed comparison of the successful and unsuccessful programme. However, I stress that my study was not a controlled experiment; it was never intended to be able to identify the impact of each individual factor on the success or otherwise of a measurement programme.

2.1 The initial introduction of measurement

Measurement was initially introduced at ES only because they were externally funded to field test the ami approach to implementing measurement [Pulford *et al* 1996]. Although this was not a particularly highly motivated start, ES quickly discovered the value of measurement and the programme was reasonably well implemented. Measurement was implemented initially only in one development team. This worked well as it meant that the measurement programme was very close to the developers but could be tightly controlled by the managers.

Measurement implementation at IS was weak. Indeed the initial motivation for a measurement programme was to provide senior managers with data to monitor productivity. Other studies have shown that weak motivations like this are commonplace [Russell 1990]. However, the relationship between weak motivations and programme failure is not proven.

A centralised Measurement Group was set up at IS to introduce measurement across the company's whole development function. There was no discernible use of established aids to measurement nor was the use of measurement piloted. The implementation strategy here seemed to consist solely of the Measurement Group instructing development departments to start collecting specified measurement data.

2.2 Developer involvement

Who designs new metrics?	Embedded Systems %	Information Systems %
Managers	20	60
Developers	0	0
Managers & developers	40	15
Don't know	35	24

Table 8.2 Participation in measurement design

Table 8.2 identifies the practitioners at each company who are most involved in designing the measures (practitioners were all asked to identify who designed measures from the given list). Although in both companies managers were very involved in the design of the measurement process, IS in particular had little developer input to the process.

Again, managers at both companies also think that developers are more involved in the design of measurement than the developers say they are. At Embedded Systems 60% of managers compared to 20% of developers think that the design of measurement is a joint effort, while at IS 27% of managers and 4% of developers think the same.

2.3 Data usefulness

Measurement usefulness	Embedded Systems %	Information Systems %
Very useful	30	15
Quite useful	60	45
Not very useful	10	25
Not useful at all	0	10
Don't know	0	6

Table 8.3: Data usefulness

Table 8.3 shows that although practitioners are generally positive about the usefulness of measurement, practitioners at ES were more positive overall than those at IS (90% of practitioners at ES and 59% at IS thought that they were useful).

	Embedded Systems %	Information Systems %
1. Is the data collected accurately?		
Accurate	40	18
Not accurate	10	41
Don't know	50	41
2. Is the right data collected?		
Yes	40	11
No	10	46
Don't know	50	43
3. Is the data manipulated?		
Often	20	27
Occasionally	40	50
Never	5	1
Don't Know	35	22

Table 8.4: Data integrity

Table 8.4 shows the confidence that practitioners had in each company's measurement programme. It confirms that the measurement programme at ES has been received more positively than the measurement programme at IS. However it also shows that most practitioners, at both companies, were convinced that the actual measurement data collected was unreliable (for example, that it was manipulated).

Table 8.5 shows a clear difference in feedback to practitioners between ES and IS. Feedback is better at ES. This probably contributes to how positive practitioners are about measurement at ES.

Measurement feedback?	Embedded Systems %	Information Systems %
Feedback is provided	30	18
Feedback is not provided	15	53
Don't know	55	29

Table 8.5: Measurement feedback

2.4 The implications of these results for companies

2.4.1 Measurement and goals

The approach adopted by each company gives me a particular insight into GQM. GQM forces an incremental approach to identifying a set of measures. It complimented ES's incremental department by department implementation strategy. This worked well for ES and they did not encounter the problems that some commentators warn of. IS, on the other hand, used neither GQM nor made use of data that was already easily available and had many problems with their measurement programme.

IS's approach to identifying a measurement set lacked a clear strategy at all. Whether GQM is being used or not there must be an explicit and legitimate use for the data collected. The use to which IS was putting its data was not clear to practitioners.

There must also be an obvious link between improvement goals and the measures that are being collected. If goals and the relationship between goals and measures are not obvious, then practitioners will not be convinced about putting effort into measurement. The validity and reliability of the data collected will, in turn, be compromised. This seemed to be happening in IS.

2.4.2 Measurement and feedback

The results also provide an interesting insight into the nature of measurement feedback. Practitioners at ES were reasonably satisfied with measurement feedback whereas practitioners at IS were not at all satisfied. This surprised senior managers at both companies. Before this study IS managers had been confident that formal feedback mechanisms were in place (though no proper evaluation of the effectiveness of these mechanisms had taken place). Managers *assuming* that they had implemented programmes effectively is also supported by the results of other studies [Neilson & Timmins 1997].

In the light of these results the managers at IS have taken steps to improve measurement feedback. On the other hand managers at ES were surprised at how relatively satisfied practitioners seemed to be with measurement feedback, as they knew that practically no formal feedback mechanisms were in place. ES did, however, use measurement on a day-to-day basis and practitioners regularly discussed measurement results informally. ES also operated an open house policy for measurement data. So although measurement results were not formally distributed or displayed, practitioners knew that they had access to the data if they wanted it. This may have encouraged practitioners to feel confident that measurement data was not being abused and, therefore, feel that they did not necessarily need to actually see the results themselves. On the other hand, I do not know whether or not the programme would have been an even greater success if more formal feedback had been provided.

2.4.3 Knowledge about measurement

There are a significant number of practitioners who *don't know* about aspects of measurement. In ES, where the measurement programme is well established and apparently successful, many practitioners do not know whether, for example, measurement data is accurate or whether the right data is collected. Practitioner ignorance is a weak link in both programmes. However, it seems to be the case that practitioners feel more strongly about some aspects of measurement than others. For instance, practitioners have very strong views on the usefulness of measurement (where there are a very small number of *don't knows*).

3. Other measurement programme criteria

In addition to the success factors discussed so far, there are a number of other criteria that are important in assessing a measurement programme. This section examines three of these criteria: how much effort is spent on measurement collection; which measurement data were actually collected; and the attitudes of developers and managers to measurement.

3.1 Measurement collection effort

There is little explicit discussion in the literature about what constitutes a reasonable overhead for a measurement programme. [Rombach *et al* 1993] suggests that 7% overall is an average effort overhead figure.

My study shows that 90% of practitioners in ES say they spend less than 3% of their working week on measurement-related activity with 10% of practitioners spending between 3 and 14% of their week on measurement (in IS: 79% spend less than 3%; 16% between 3 and 14%; 4% between 14 and 29%). Although neither programme appears to involve a big effort overhead, practitioners at ES actually spend less time collecting measurement data than practitioners at IS (probably because the former use automated tools).

Ironically, IS has not been as effective at implementing measurement as ES, and yet is spending more effort on its measurement programme. IS is also collecting more measurement data. It seems likely that some of the data collected by IS are not needed and are probably not actually used (the literature warns against this [Baker 1991]). It is also suggests that ES has been more successful that IS in distinguishing the small number of useful measures from all the possible measures that could be implemented [Hetzel 1993].

3.2 The measurement data collected

There has been no comprehensive international survey that quantifies industrial penetration of measurement since [Drummond-Tyler 1989]. Indeed there appears to be disagreement amongst influential software commentators about the extent to which measurement has penetrated the software industry. For example, [Jones 1991] claims an encouraging industry-wide take up of measures such as function points, while [Hetzel 1993] estimates a much lower figure. One of my aims was to try and quantify the extent to which different measurement data was actually collected. A summary of my findings appears in Table 8.6.

Measures	Total				Manager				Developer			
	Question One		Question Two		Question One		Question Two		Question One		Question Two	
	ES	IS	ES	IS	ES	IS	ES	IS	ES	IS	ES	IS
Function points♦	-	86	-	15	-	87	-	19	-	85	-	11
Measurement for size estimates*	85	50	40	11	90	57	60	13	80	44	20	9
Measurement for cost estimates*	90	80	30	26	90	85	50	56	90	76	10	7
Measurement for effort estimates*	70	65	35	27	90	77	60	42	50	54	10	13
Analysis inspection data	40	35	10	8	50	41	20	13	30	30	0	4
Design review data	75	39	25	8	80	44	50	10	70	34	0	6
Design effort data	40	31	20	9	50	40	30	11	30	23	10	7
Code interface data	65	16	20	5	60	13	30	4	70	19	10	6
Code complexity data	70	7	20	0	90	8	30	0	50	6	10	0
Lines of code data	80	43	25	12	100	52	40	19	60	35	10	7
Coding effort data	45	20	10	12	70	49	20	19	20	26	0	6
Code inspection data	30	29	15	7	40	38	20	10	20	22	10	4
Fault rates	25	14	10	2	20	15	10	4	30	13	10	0
Defect densities	20	15	10	3	30	23	20	4	10	9	0	2
Change data	60	28	20	5	70	32	30	6	50	24	10	4
Testing effort data	40	31	10	7	60	49	20	13	20	15	0	2
Test review data	65	25	20	6	70	34	30	11	60	17	10	2

Key: Each entry represents the percentage of respondents answering *yes* to the following questions:

Question One - '*Does your company collect the following measurement data?*'

Question Two - '*Do you know, from* <u>*personal involvement*</u>*, that the following measurement data is collected?*'⊥

Table 8.6: The measures collected

The table lists a sub-set of the measures contained in both companies official measurement programmes. Significantly the IS full list of official measures is considerably longer than ES.

I analyse measurement activity in terms of the knowledge that practitioners have about what is being measured generally, and in terms of the personal involvement of practitioners in data collection. My rationale is that, although there may be an *official* measurement programme in place, unless practitioners are aware of that programme and actively involved in that programme, then the company has not created the necessary measurement culture for the programme to be effective. Indeed I found that many of the measures cited by both companies as part of their 'official' programme were so little known about and used, that it is difficult to accept that those measures are actually part of real measurement programmes.

♦ IS were very keen on function points, so I asked them separately about function point penetration.

* I asked only about fairly general measurement in order to avoid fragmenting the results.

⊥ Strictly speaking this paraphrases what they were asked.

3.2.1 Measurement awareness

Table 8.6 shows that there is a consistently higher level of measurement activity at ES among both managers and developers. At IS, despite the official measurement programme containing lots of measures, awareness of these measures is generally low.

Although there is a higher general level of awareness at ES, the awareness gap between managers and developers is lower at IS (an average 13 percentage point difference between managers' and developers' measurement awareness levels at IS and a 20 percentage point difference at ES). Lines of Code measurement are a good illustration of this awareness gap: 100% of managers and 60% of developers at ES know that lines of code data is collected (an awareness gap of 40 percentage points), compared to just 52% of managers and 35% of developers at IS (an awareness gap of 17 percentage points).

Throughout this study I found that managers tended to be more bullish than developers about how their company is performing and this could explain why managers claim to be more knowledgeable about the data collected. However, in this case, it is likely that managers have more access to information than developers and are, therefore, actually more knowledgeable about what measures are in the official measurement programme.

Generally practitioners at both companies exhibit poor awareness of what is happening to review and inspection data. Both companies use reviews and inspections regularly, and the review process can be a rich source of measurement data. Table 8.6 shows that the use of this data is not obvious to managers or developers at either company. This suggests that the data is being used sub-optimally. This is probably a weakness of many programmes and is another example of companies not putting into practice the published experiential advice.

3.2.2 Involvement in data collection

Table 8.6 also shows that managers at both companies are more personally involved in measurement data collection than developers, though ES still has a higher overall participation level than IS. Also, while ES has a successful measurement programme, there is minimal developer involvement in data collection. This is because the measurement programme at ES is deliberately driven by managers and it makes use of automated data collection. On the other hand, the measurement programme at IS seems generally inactive, with few managers or developers directly participating in data collection. Indeed only 19% of managers and 7% of non-managers at IS say that they know from personal involvement that LOC data is collected (IS does not use any special purpose automated data collection tools and is a largely paper-based programme).

These results suggest that it is important practitioners know what is going on in terms of measurement, but that personal involvement in collection is not very important. This finding supports other experiential reports [Pfleeger 1993] [Daskalantonakis 1992] where using automated tools and keeping practitioners appraised of what is going on were found to be some of the pre-requisites to measurement success.

Table 8.6 also shows that while ES has the most active measurement programme it is certainly not perfect. ES has the following particular measurement problems:

- **Poor transparency**
  Although 70% of managers say that coding effort data is collected, only 20% of developers know that it is, and no developers think that they are involved in collecting it. It is difficult to understand how accurate lifecycle effort data can be collected without the participation of developers (although they may not realise that time sheets are used for that purpose). It is also difficult to understand the purpose of collecting such data if developers do not know that it is required. It cannot be used to motivate productivity if people do not know that it is being collected.

- **Poor goal-measure coupling**
  Although 90% of managers say that code complexity data is collected, only 50% of developers know that it is collected and very few managers or developers are involved in collecting it. The main purpose of collecting complexity data is to keep control of complexity. If developers do not know that complexity data is being collected then they are unlikely to take reducing complexity seriously. This makes collecting complexity data ineffective. This shows that ES needs to collect data in a goal orientated way and be very clear about why they are collecting specific data.

3.2.3 The core set of measures

Embedded Systems	**Information Systems**
Measures for resource estimates	Measures for resource estimates
Lines of Code	Function points
Design review data	Lines of code
Code complexity data	

Table 8.7: Core measurement

Table 8.7 gives an indication of the most frequently used measures in each measurement programme (this has been constructed from the activity levels given in Table 8.6). Although ES is generally more active in its measurement activity, both companies favour a stereotypical set of core measures. Indeed there seems little rigor in the way measures have been choosen, despite the fact that the measures choosen can have a significant impact on the whole process [Pfleeger & McGowan 1990]. Both sets of measures are dominated by size and effort measures (primarily used for resource estimation and productivity measurement), rather than measures that can measure quality. This result supports other research such as METKIT [Russell 1990] which discovered the same company emphases. The low priority given to quality measurement runs contrary to the advice of all commentators. For example, the Grady & Caswell book and the ami handbook both emphasise the need to collect product quality measures. Over-emphasis on cost orientated data is another common fault in measurement programmes.

3.3 Developer attitudes to measurement

One of the most important features of a successful measurement programme is how positive developers are about the use of measurement. Successful programmes are characterised by minimal developer resistance. Failing to generate positive developer attitudes is likely to seriously undermine a programme's chance of success.

3.3.1 Managers' and developers' attitudes

Measurement usefulness	Embedded Systems		Information Systems	
	Managers %	Developers %	Managers %	Developers %
Very useful	40	20	22	9
Quite useful	60	60	49	41
Not very useful	0	20	19	30
Not useful at all	0	0	9	11
Don't know	0	0	2	0

Table 8.8: Managers' and developers' views of measurement

Table 8.8 shows the difference in practitioner attitudes between IS and ES. It shows that developers and managers are more positive about the use of measurement in ES than practitioners in IS.

However, a significant influence on attitude towards measurement was job seniority. In both companies managers were much more positive about the use of measurement and about how well measurement had been introduced and managed. Furthermore the more senior a manager the more enthusiastic about using measurement they were. For example, 87% of senior managers at IS were positive about measurement compared to 40% of middle managers and 46% of developers. The pattern was similar at IS. This suggests that measurement has been implemented in a way which actually makes it more useful to managers. My findings on this issue are analogous to Weinberg's CMM experiment. He found that the more senior someone is the more likely they are to say that key CMM practices are in place [Weinberg 1993]. An Australian study found a similar discrepency [Wilson *et al* 1995].

However, in this study I found that developers were more positive about measurement than conventional wisdom has led me to believe. It is generally thought, especially by managers, that developers are unenthusiastic about QA mechanisms like measurement, and that developers cannot see the value of using them [Nicholson *et al* 1995]. My results actually show that 80% of developers at ES and 50% of developers at IS were positive about the use of measurement. I think that improving practitioners' perceptions of measurement will significantly improve a programme's chance of success.

It has been said that when developers are asked what they think about software QA mechanisms they rationalise that such things are useful. However, it has also been said that the reality of getting developers to participate in quality initiatives is a different story, as developers can find many reasons why their particular work must be exempt [Sommerville 1996]. This argument could be made to explain the positive attitudes to measurement that I found in this study. However I believe that even if this argument is true (and usually no one actually asks developers what they do think about such initiatives) and developers are only theoretically positive about measurement, then companies need to work towards releasing this positive potential. The suggested relationship between positive perceptions and negative action needs more research.

3.3.2 Perceptions of integrity

Table 8.8 also strongly supports my view that practitioners' perceptions of measurement are strongly influenced by the reality of their measurement programme rather than any preconceived notions of measurement being good or bad. If this were not the case I would expect developer and manager views to be more aligned between the two companies. In fact, Table 8.8 shows that practitioner perceptions vary significantly between the two companies, even though within each company the perception patterns of managers and developers are very similar. This has important implications for managers as it means that what practitioners think about measurement and, therefore, how practitioners respond to measurement, is within managers' control. Too frequently managers assume that developers will be negative about measurement *per se*. My results suggest that developers respond to the measurement experiences that they actually have.

Table 8.9 illustrates that the integrity of measurement data seems to have a powerful influence on the perceptions that all practitioners have about measurement. Managers at both companies are significantly more convinced than developers that: the data collected is accurate; enough measurement feedback is provided; the right measurement data is collected.

Such a manager/developer gap is worrying. In an effective measurement programme the measurement data not only has integrity, but developers believe it has. My study has not examined the integrity of the measurement data at the companies, so I do not know whether the data has integrity or not. What I do know, however, is that many of the practitioners affected by measurement do not believe that the data has integrity. This perception will probably do as much damage to the measurement programmes as the data really having no integrity. In fact this relationship is illustrated in Tables 8.8 and 8.9 as developers at ES are less negative about the integrity of measurement data, and ES developers are also more positive about using measurement.

	Embedded Systems		Information Systems	
	Manager	Developer	Manager	Developer
1. Is the data collected accurately?	%	%	%	%
Accurate	60	20	31	8
Not accurate	10	10	39	43
Don't know	30	70	29	50
2. Measurement feedback?				
Feedback is provided	50	10	25	11
Feedback is not provided	20	10	56	50
Don't know	30	80	19	39
3. Is the right data collected?				
Yes	60	20	13	10
No	0	20	48	44
Don't know	40	60	39	44
4. Is the data manipulated?				
Often	20	20	29	26
Occasionally	30	50	58	41
Never	10	0	0	2
Don't know	40	30	13	31
5. Who manipulates the data?				
Developers	0	10	21	9
Managers	40	30	42	39
Neither	10	0	2	2
Both	20	20	24	20
Don't know	30	40	10	30

Table 8.9: Managers and developers views on measurement efficacy

3.3.3 Developer and manager involvement

There are two probable explanations for the manager/developer gap that I observed. First, it is probably the case that managers did have more access to measurement information and data. So, for example, there probably was enough feedback to managers, and managers probably were in a better position to judge that the data was collected accurately. This is supported by the fact that a very small number of developers at each company thought that software measurement data was being used to judge individual performance and to affect salaries. Both of these views were inaccurate as neither company did use its data in that way (or admitted to using it in that way). Second, managers in both companies were more actively involved in setting up and managing the measurement programmes. Consequently managers are probably less likely to criticise something that they have been instrumental in setting up. This may be further compounded by the fact that no evaluations of measurement had taken place at either company, and so managers may have been unaware of the problems that I uncovered.

4. An epilogue

These results were taken very seriously by the measurement managers at IS. However, it is unfortunate that IS had not previously evaluated their programme themselves. One year on from this study, many of the weaknesses that I identified in the measurement programme either have, or are in the process of being, rectified. The programme has been radically improved. The impetus for the improvement has come from this study and from IS having since been taken over by a company with a more established QA and measurement regime. In particular the programme has been improved in the following ways:

- **Devolution.**
  The programme has been made much more local to developers, with measurement responsibility now having been devolved to individual development teams.

- **Support.**
  The centralised Measurement Group now acts in an advisory rather than managerial role.

- **Structure.**
  Managers have made a big effort to improve transparency and feedback within the programme.

- **Focus.**
  The set of measures has been revised and is now smaller and more goal focused. The data now collected specifically address the areas of: effort, size, changes, defects and duration.

The measurement programme at ES continues to be carefully managed and improved in an incremental way. The measurement set has been refined and some measures are no longer collected. The programme is viewed by ES senior managers as so successful, that it is being 'rolled-out' to all other software teams in the company.

Neither company was, however, prepared to divulge to me measurement data about how the quality of their software has changed during the study. This means that even at this stage in the study it is impossible for me to report on how the quality of software produced by ES compares to the quality of the software produced by IS, nor how quality has changed while the measurement programme has been live.

5. Conclusions

5.1 General findings

In assessing the programmes in these two companies it is worth noting that the the way in which each initally implemented measurement may be significant to long term success. ES basically piloted measurement in one small and focused department. In this environment it is more likely that most people in that department will be aware of measurement and have some role in it. It could also be argued that within the limited confines of this concentrated effort measurement is highly likely to be successful. Successfully transferring measurement to the rest of the company without such *hothousing* will be the difficulty - however many commentators (for example, DeMarco 1982) believe that successful programmes start small and grow incrementally.

In contrast IS deployed measurement throughout the whole company. Although this programme was not obviously successful nor was it failing. It could be argued that such a comprehensive programme is likely to take longer to optimise, but that given time a programme that survives long term is more likely. Furthermore, it could also be argued that IS's programme is currently more successful than ES's as its penetration throughout the company was much more comprehensive. Both programmes need to be monitored in the long term to answer these important questions.

However, not withstanding these issues regarding the nature of the companies' initial measurement programmes, I generally found that the consensus 'success' factors in the literature are generally correct, but that this advice is not often heeded. Success seemed particularly linked to a the following factors:

- **Research.**
  Doing background research on other measurement programmes and following the advice given in the published experiential reports;
- **Participation.**
  Involving developers in designing the measurement programme and keeping them informed of the progress of the programme;
- **Incremental implementation.**
  Using an incremental approach to implementation, and initially piloting the measurement programme;
- **Sensitive management.**
  Acknowledging and being sensitive to the concerns that developers have about the use of measurement data.

I also believe that both measurement programmes would have benefited from having been evaluated earlier and more regularly.

In the two case study companies I found a discrepancy between the 'official' measurement programme and the actual measurement programme. Many of the measures that were formally part of the 'official' programme were being used so infrequently and by so few practitioners that they were not part of the programme in any real sense. This suggests that although inappropriate measures might be implemented they will not actually be used within the programme. This means that measures need to be constantly evaluated for usefulness. If measures are not generating useful data then effort is wasted by keeping them in the programme. This study showed that at IS, managers often were out of touch with the 'real' measurement programme.

I found that the core measures in both companies were based around measuring effort and size. This confirms emphasis on resource and productivity information at the expense of measuring quality. Little effort was devoted to collecting fault data.

I also found that managers were more positive about measurement than developers. Furthermore, managers were much more positive about programme integrity. Measurement positiveness was generally related to seniority - the more senior a practitioner was, the more positively they perceived measurement. However, I did find that developers were more positive about measurement than conventional wisdom leads me to believe (although neither company in the study was very successful at exploiting this positive potential). My results suggest that improving practitioner perceptions of measurement should be a priority for companies implementing measurement. Indeed there is now mounting evidence to show that the more positively developers perceive measurement, the less resistance there will be, and the more likely the measurement programme is to be successful.

In finally contemplating the two measurement programmes I suspect that one of the most important, but intangible success factors was the approach and attitude of measurement programme managers. The successful programme was managed with a tenacious commitment to see measurement work. In contrast, the unsuccessful programme had a half-hearted feel about it, despite the fact that it was a very ambitious programme that was expensive to set up. Indeed at the outset the odds were probably stacked against the successful company: it had a weaker QA framework and lower staff morale. This suggests that companies implementing measurement not only need to make use of the good practice discussed in this and other work, but also must manage those programmes with commitment and enthusiasm. Indeed [Karolak 1993] confirms that successful programmes require such management vision.

5.2 Evidence relating to the study hypotheses

5.2.1 Hypotheses about the company

i. Measurement is most successful when a quality culture exists within the company.

I could find no convincing evidence to support this hypothesis. IS appeared to have more of a quality culture in terms of how much feedback and so on practitioners received (see Seection Three in Chapter Seven) and yet had a less obviously successful mesurement programme. Furthermore, ES had quality culture that appeared weak and yet seemed to have a more successful measurement programme. The relationship between a company's culture and the effectiveness of measurement needs further work.

ii. It is common for companies not to practice what they claim to in terms of measurement.

Table 8.6 in section 3.2 provides some evidence in support of this hypothesis. It shows data suggesting that many measures claimed as part of mesurement programmes were not collected and used.

5.2.2 Hypotheses about the practitioners

i. Practitioners' attitudes affect the success of measurement.

Table 8.3 in section 2.3 provides some evidence in support of this hypothesis. It shows that practitioners in ES were much more positive about measurement than practitioners in IS. Given that ES seemed to have the more successful measurement programme then one interpretation of this result is that positive attitudes improve measurement programmes. However, again, more research is needed to establish the cause and effect of this relationship. Do positive attitudes affect the quality of measurement programmes or does the quality of measurement programmes affect attitudes?

ii. Managers and developers have different views on which measurement practices are worthwhile.

Section 3.3 provides some evidence in support of this hypothesis. It shows that generally managers value measurement more than developers do. Furthermore, section 3.1 provides evidence that shows managers are much more active in measurement than developers (though this evidence is not directly related to the hypothesis).

iii. Managers are enthusiastic about measurement.

Section 3.3 provides further strong evidence in support of this hypothesis. It shows that not only are managers more enthusiastic about measurement, but that enthusiasm increases with management seniority.

iv. *Developers resist the implementation of measurement.*

Section 3.3 also adds further evidence in support of this hypothesis. It shows that although developers do not seem to resist measurement they are not as enthusiastic about it than mangers.

5.2.3 Hypotheses about the implementation process

i. *The goals of measurement must be clear to practitioners if the programme is to succeed.*

I could not find evidence to support this hypothesis. Section 2.2 in Chapter Seven shows that in neither company did practitioners feel that company goals were made clear to them. Furthermore, in the same section I also show that there was significant company/practitioner goal incongruence in ES. There was less goal incongruence in IS. However, ES's measurement programme was considered successful. It is, therefore, unclear what effect goal clarity had on the programme.

ii. *Effective measurement must be transparent to practitioners if the programme is to succeed.*

Generally the measurement process seemed more transparent in ES than IS, so the study shows some evidence in support of this hypothesis. Section 2.3 shows that practitioners in ES were generally more knowledgable than practitioners in IS about the measurement process.

iii. *The usefulness of measurement should be obvious to all practitioners if the programme is to succeed.*

I found evidence that in ES's successful programme measurement was considered much more useful than in IS where the programme was less successful (see section 2.3).

iv. *Developers should participate in designing measurement if the programme is to succeed.*

I found some evidence in support of this hypothesis. Section 2.2 shows that there was a generally higher level of participation in measurement in ES where the measurement programme was more successful. In IS's less successful programme participation was lower.

v. *Measurement initiatives are likely to be most successful when implemented in a bottom-up, incremental way.*

It proved impossible to collect data to test this hypothesis as none of the companies in this study implemented QA or measurement in anything other than a top-down way.

vi. *Practitioners should have confidence in the integrity of measurement data if the programme is to succeed.*

I found evidence broadly in support of this hypothesis. Section 2.3 shows that practitioners in ES had more confidence in measurement data than practitioners in IS.

vii. Feedback on measurement needs to be provided if the programme is to be successful.

The evidence in support of this hypothesis was inconclusive. Although section 2.3 shows that ES generally had better feedback on measurement than IS, the levels of feedback were not high. Again, more research is needed to confirm the relationship between feedback and measurement success.

viii.Any perceptions that data is being used inappropriately will make practitioners negative about measurement.

I found strong evidence in support of this hypothesis. Section 2.3 shows that pratitioners in IS were very unsure whether measurement data were being used approapriately. Many practitioners believed that data was manipulated. Practitioners were generally less worried about the way data was being used in ES.

Chapter Nine

Managers' and developers' experiences of quality

1. Introduction

In this chapter I report on the final phase of my study. During this phase I followed-up more of the QA issues that emerged earlier in the study. I particularly focus on:

1. The attitudes of developers and managers to quality and QA.
2. Quality certification in ordinary companies.

Some of the companies in this phase of the study were quality certified and some were not. This finally gave me an opportunity to explore some of the certification issues that emerged during the reconnaissance study reported in Chapter Five.

In this part of the study I applied the Repertory Grid Technique [Fransella & Bannister 1977] to practitioners in five different companies. Using this technique I generated exceptionally high quality data about the QA experiences and views of managers and developers. This technique, how I used it and the companies to which I applied it, are described in the Study Methods chapter (Chapter Two).

In this phase of the study I further address many of the hypotheses presented in Chapter One. Again I add further evidence to some already examined and examine others for the first time. I summarise how the data presented in this chapter supports the study hypotheses in the conclusions section.

The chapter is structured as follows: Section Two looks at the industry-wide issues that emerged in this phase of the study. Section Three presents those issues related to particular companies. In Section Four I discuss quality certification in the companies. Section Five considers other factors influencing quality and QA in the companies. The conclusions of this phase of the study can be found in Section Six.

2. Industry-wide issues

This section presents the issues that arose in the group sessions across all the companies in the study. Some issues only arose in the manager sessions; other issues only in developer sessions, but some issues arose in both manager and developer sessions.

2.1 Manager and developer issues

2.1.1 Positive about quality improvement

All the practitioners in this study had positive attitudes towards QA and all were keen to see software quality in their company improved. All also said they were amenable to ways that improvement could be achieved. Group norms about quality were also quick to emerge at each company. These norms were generally company-specific and dependent on a company's particular approach to QA.

2.1.2 Growth in formalism

In all but one of the companies formal QA in the software development process was perceived to be increasing, (this was not the case in company B and the reasons for this are discussed later). This increased formalism included more standards, procedures, methods, measurement and so on. On the whole practitioners were positive about this growth in formalism. They were particularly positive about the formalism that reflected and disseminated existing good practice. Indeed, developers in company C said that their QA procedures had simply formalised the way in which they knew they should be doing things anyway. Practitioners generally felt that the formalism they had been involved in developing, and which was implemented slowly and sensitively, did not meet with resistance.

2.1.3 The impact of experience

Practitioners with the most experience of working within QA programmes were generally most positive about formal QA. Ignorance about what it is to work within a QA programme seemed to make practitioners fearful and resistant.

2.1.4 Pressures on quality

Managers and developers were both concerned about the impact tight deadlines and diminishing resources had on quality. There was a strong feeling that the current commercial climate encouraged companies to take a short-term view of quality. Practitioners thought that most companies (and their customers) considered the quickest and cheapest solution to be the best solution. Practitioners were particularly frustrated that tight schedules led to mistakes which compromised quality. They also said that formal QA *went out of the window* when deadlines loomed, and that, in the long-run, this led to re-work. Excessive schedule pressure seems to damage quality on a world-wide basis [Jones 1996].

In some companies practitioners felt that they also worked under an implicit job threat. In these companies practitioners believed that if they could not deliver software on time then they would be replaced by practitioners who could. These problems were particularly acute in company D where practitioners said that senior management insisted that software was delivered on time, believing that quality problems should be sorted out after delivery. Practitioners' general perception was that despite companies saying that quality was important, companies actually valued deadlines more highly than quality.

2.1.5 The need for flexibility

All practitioners stressed the need for flexibility within QA programmes. Practitioners in the companies with mature QA programmes stressed how the success of those programmes depended on the flexibility inherent in them. Practitioners in the companies with lower QA maturity believed that QA programmes were not able to be flexible. Practitioner perceptions of QA flexibility were related to their levels of QA experience. Many practitioners who had experienced mature QA programmes said that they had been surprised at how flexible those programmes could be in reality (they too had expected them to be more rigid). Many practitioners were quite ignorant about how unprescriptive QA programmes can be.

2.1.6 The dangers of bureaucracy

Practitioners all commented on the dangers of introducing unproductive bureaucracy in the name of QA. Again views fell into two camps: those practitioners with experience of QA programmes and those without. Most practitioners with experience of working in an environment with a QA programme said that, although they had worried about the prospect of excessive bureaucracy, their programme had not proved to be full of 'red-tape'. However, one or two practitioners had experienced highly bureaucratic QA programmes that had made them very negative about QA programmes generally.

Practitioners without experience of QA programmes were much more concerned about bureaucracy. Many were convinced that the smallest task would become a bureaucratic nightmare. This perception was particularly acute in company B, where practitioners seemed to have a pathological hatred of anything to do with formal QA. This was taken to the extreme of a senior manager having been assigned the task of 'running interference' on a major customer's QA programme.

Practitioners in most of the companies felt that some companies had taken formal QA too far, and that this had scared some companies off QA altogether.

Table 9.1 provides a summary of the spread of the general issues discussed above across the companies in this part of the study.

Issues	Issue occurrences across the companies A	B	C	D1	D2
1. Positive about quality improvement	✓	✓	✓	✓	✓
2. Growth in formalism	✓	×	✓	✓	✓
3. The impact of experience	✓	×	✓	✓	×
4. Pressures on quality	✓	×	✓	✓	✓
5. The need for flexibility	✓	✓	✓	✓	×
6. The dangers of bureaucracy	✓	✓	×	✓	×

Table 9.1: Industry-wide issues across the companies

2.2 Middle manager issues

2.2.1 Top-down pressures

Middle management generally felt under pressure from senior management to improve quality. However these managers also felt that senior managers did not fully understand the difficulties of software development. This was a particular issue in companies whose primary product was not software. In some companies such as D, managers felt that senior management had a 'band wagon' approach to improving quality. Managers in company D described how senior management had half-heartedly implemented a sequence of different quality silver bullets (most of which were subsequently abandoned). Managers in one or two of the companies were concerned that the approach of senior management had made practitioners cynical about all quality initiatives. There was a general perception that, although senior management were keen to be seen implementing quality improvement initiatives, they were only really interested in software being delivered on time and in budget. Middle managers believed that delivering high quality products was a secondary concern of senior managers; delivery on time was perceived to be much more important to senior managers. There was a general feeling that senior managers only really paid lip service to quality.

2.2.2 Bottom-up pressures

Middle managers also felt that they were under pressure from developers about quality. Managers in three companies said that developers were keen to see software development improved and actively advocated ways of achieving improvement. Managers also said that grassroots quality improvements were most long lasting and successful.

2.2.3 Quality empowerment

A recurrent theme in sessions with managers was their attempts to create a quality culture. Managers said that their priority was to create a grassroots quality culture where individual developers took responsibility for the quality of their software. In some companies (company A in particular) managers emphasised the importance of a 'no-blame' ethos. Managers in company A were trying to create a culture where people could be open about their mistakes so that quality could be improved in the long term. Managers also thought that professionalism was highly significant to software quality. This meant that managers viewed developers as responsible for the quality of the software they personally produced. Indeed managers and developers both felt that quality software was ultimately dependent on developers.

2.2.4 Quality infrastructure

Managers in some of the uncertified companies felt strongly that it was important to get the QA infrastructure right before thinking about quality certification. They were aiming to develop a QA programme which first and foremost improved quality, and only when that was working effectively think about having the programme certified. They thought that too many companies had gained quality certification with minimal procedural change and no real quality culture. They also speculated that such companies did not have QA programmes which actually improved quality.

2.2.5 Information and guidance

Managers felt strongly that there was not enough information and guidance on implementing QA programmes (a result also supported by the findings of other studies [Bache & Neil 1995]). This was a particular issue in the two companies (A and C) with the highest QA maturity. Managers in company A felt that this lack of proper information and support had initially led them to implement an overzealous QA programme in their attempt to be sure of achieving certification.

Table 9.2 provides a summary of the spread of the manager issues across the companies.

	Issue occurrences across the companies				
Issues	A	B	C	D1	D2
1. Top down pressure	✗	✗	✓	✓	✓
2. Bottom up pressure	✓	✗	✓	✓	✗
3. Quality empowerment	✓	✗	✓	✗	✗
4. Quality infrastructure	✓	✗	✓	✗	✓
5. Information and guidance	✓	✗	✓	✗	✗

Table 9.2: Manager issues across the companies

2.3 Developer issues

2.3.1 Increased formalism

Developers in almost all of the companies said they wanted a more formal approach to quality. Exceptions to this were: company B, where an anti-quality culture existed, and company A, which already had a certified quality programme and where developers were generally happy with existing levels of formality. Developers in the companies with the least formal QA were most vocal in their desire for more. Developers' craving for more formality was illustrated when developers in company D were positive about their company recently introducing timesheets.

Developers said they wanted more formal QA because they felt that:

- They wanted to know the 'right' way of doing tasks.

- Explicit standards and procedures that would enable them to show managers exactly what is involved in producing a new software system. They felt this would put them in a stronger position to argue for realistic schedules and resources. Developers in company C described formalism as 'a good stick to beat managers with' (indeed this has been shown to be a direct result in Boeing [Lytz 1995] and has been cited as a benefit in other programmes such as the US Navy [McGarry 1996]).

- If things went wrong they would have more protection from blame if they could show they had followed the appropriate procedures.

- New staff could be integrated into the company more easily.

Developers said they were disappointed at how slow companies are to formalise and disseminate good practice. There was disappointment in company D that senior management were not keen to nurture the grassroots quality initiatives that developers valued. Developers in D thought that their senior managers were keen on formal QA that had little real value. They also believed that senior managers had a 'band wagon' approach to formal QA and this led to dogma-driven initiatives. This problem is also reported in other studies of companies where 'politically' motivated initiatives that fail to address real needs are handed down from 'on high' [Neilson & Timmins 1997].

Developers in company D also said that they would like to see a more formal approach taken to senior management planning and decision-making. The current decision-making process was perceived to lack transparency and seemed *ad hoc*.

The general developer consensus was that although developers were keen on formal QA, they were not keen on the superficial QA that senior managers usually introduced.

2.3.2 Appropriate levels of detail

Developers felt that it was important to be aware that formalism can quickly become outdated and that a QA programme needs constant maintenance. Developers felt that maintenance was often neglected and this led to QA programmes degrading.

2.3.3 Quality programme evasion

Developers felt that some of the procedures within QA programmes were too easy to evade (including developers in the companies with most QA maturity). Indeed there was a general feeling that when 'the going got tough' managers encouraged QA evasion.

2.3.4 A threat to creativity

Very few developers in the study mentioned conflicts between formal QA and creativity. Only in company B was it considered an issue, although one manager in company C mentioned it as relevant to R&D developers.

Table 9.3 provides a summary of the spread of the developer issues discussed above across the companies in this part of the study.

	Issue occurrences across the companies				
Issues	A	B	C	D1	D2
1. More formalism	✗	✗	✓	✓	✓
2. Appropriate levels of detail	✓	✓	✗	✗	✗
3. Quality programme evasion	✓	✓	✓	✓	✗
4. A threat to creativity	✗	✓	✓	✗	✗

Table 9.3: Developer issues across the companies

3. Company-specific issues

3.1 Opposition to quality

Company B had a culture which was diametrically opposed to any formal QA. This company was a classic CMM level one company. It was a small software house headed by a charismatic managing director who believed that quality depended only on the calibre of the developer (it could be argued that in this view he is not alone [Curtis *et al* 1995]). He claimed to produce high quality software because he only employed highly creative able developers who were not restricted by bureaucracy (he specifically recruited physics rather than computing graduates).

During the study I did, however, come across anecdotal evidence suggesting that the quality of the software produced by this company was not high. However company B had the type of client profile reported by [Neilson & Timmins 1997] where they were convinced that they would retain their major customers regardless of a QA programme.

The managing director felt that QA programmes depress and de-motivate creative people. Practitioners in this company were vitriolic in their opposition to formal QA, perceiving it only as a restriction on their creative freedom. They thought that QA programmes did not improve quality, but instead created work in QA departments for failed developers and empire building managers. They also believed that quality certificates were pointless, as they are awarded on the basis of consistency rather than quality.

According to organisational theory company B is a classic *random* organisation. It had light and flexible management with many informal and undisciplined ways of working. It relied on personal persuasion and influence, and it had a culture that empowered *individuals*. Such organisations are renowned for recruiting technical 'stars' who influence others [Constantine & Lockwood 1994].

3.2 Damaging senior management styles

Quality initiatives in company D2 were overshadowed by a hard-line senior management style and a highly political company. Practitioners felt that a 'watch your back' way of working was necessary. They felt that the environment made improving quality difficult and made grassroots quality initiatives risky for individuals. Indeed this organisation seemed to be a classic *closed* organisation [Constantine and Lockwood 1994], where a hierarchical and disempowering organisational culture existed.

Practitioners felt that senior managers were keen for the company to be seen implementing the latest quality initiative. This left practitioners lurching from one half-hearted quality initiative to the next. The result was a disillusioned and frustrated IT section who could see many quality problems but felt powerless to address them (again [Neilson & Timmins 1997] report such senior management attitudes and approaches). This company also illustrates the need to consider QA beyond face value. Indeed, [Wilkinson & Witcher 1993] report that for effective quality initiatives the political processes within an organisation must be explicitly accounted for.

3.3 Outsourcing

An important influence on the quality views of practitioners in company D1 was outsourcing. A major part of their department's work had been outsourced and this led to an interesting quality juxtaposition. Not only were practitioners developers of software, but they were also customers of another software development company. As customers they were unhappy with the quality of the software that they were receiving, but felt powerless about the quality of their outsourced software. This made practitioners very keen on formal QA for their software suppliers, and they advocated quality certification for their supply company (though they were less keen on quality certification for themselves).

4. Quality certification

4.1 Motivation

Generally most practitioners were cautiously positive about quality certification and worried about bureaucracy and flexibility. Overall practitioners seemed to be torn between wanting the discipline of a certified QA programme and wanting plenty of flexibility within the programme. Practitioners felt that although certification *per se* did not improve quality, the process of certification had motivated many companies to take quality seriously. However some practitioners believed that it was more important to put company QA structures in place before starting to think about quality certification. These practitioners said that too many companies only implemented a QA programme to gain quality certification. A minority of practitioners, predominantly in company B, thought that quality certification was a 'bandwagon' with nothing to recommend it.

4.2 Knowledge

There was a great deal of ignorance about how prescriptive certified QA programmes need to be. Practitioners with no experience of certified QA programmes had an exaggerated notion of how rigid and bureaucratic those programmes were. Practitioners who had experienced certified QA programmes were, on the whole, positive about those programmes. However two practitioners were very negative about their experience of certified QA programmes at previous workplaces. They recounted how these QA programmes had been very heavy handed and they felt put off certification because of this.

4.3 Effect

Practitioners within the certified companies thought that software quality had improved since certification (although no company had measured this perceived improvement). On the other hand, practitioners in the uncertified companies all voiced varying degrees of scepticism as to whether certification really could improve quality. In particular practitioners were concerned that certification only meant that software was developed consistently rather than of high quality. Many practitioners believed that the resources used on gaining certification would have a bigger impact on quality if used to simply improve the development process. Practitioners at D believed the reason they did not have certification was simply that senior management did not want to use resources on certification (although the company had done so to certify a publicly visible aspect of its service). Senior management's approach to certification in company D led me to suspect that certification was being used mainly as a marketing tool in that company.

4.4 Satisfaction

Company C seemed to be most satisfied with their approach to certification. Although software at C was not specifically certified, ISO9001 principles had been applied to software development within the company-wide certificate. Practitioners felt that they had the best of all worlds with their minimalist but helpful QA programme. They felt they had useful QA structures without a massive bureaucracy. Furthermore practitioners felt that the QA programme in C had been implemented with staff consultation and input.

Positive comments on software quality certification included that it:

- looked good to customers;
- improved the status of IT within the company;
- motivated staff to do things correctly;
- helped new people get up to speed.

Negative comments on certification included that it is:

- irrelevant to quality;
- less important than having a QA infrastructure and culture;
- about producing software consistently not necessarily of high quality;
- nothing without practitioner professionalism and ability;
- *"more about putting shelves on the wall to house the paperwork than anything of real value"* (manager in company D2);
- *"a guarantee of quality despite incompetence"* (manager in company B);
- *"for companies that employ monkeys"* (manager in company B);
- *"A way for certification bodies to hold small companies to ransom"* (developer in B).

Particular certification problems which company A experienced during TickIT certification related to:

- **Information**
  They experienced a lack of information and guidance about what is really necessary to gain certification and what is over the top ([Bache & Neil 1995] also report this).

- **Consultants**
  They experienced poor performance from their quality consultants. They found that some consultants did not know enough about certification and their advice was positively damaging. This is interesting as Company A's experience contradicts the published advice where companies are encouraged to use gurus and promoters. On the other hand this company was very complimentary about the help and advice that they had received from their TickIT auditors. This shows that experience with third parties seems mixed and depends totally on the calibre of the individual(s) concerned.

5. Other quality and QA factors

5.1 Company factors

5.1.1 Culture

Although there was broad consensus between developers and managers across the companies in this study, each company did have its own particular quality culture. The biggest single influence on a company's quality culture appeared to be senior management's attitudes to quality. Senior management seemed to set the quality tone in every company. Practitioners, on the whole, simply responded according to the organisational culture in which they found themselves. This finding is supported by motivation theory which describes how humans are motivated: workers consider themselves as one unit in co-ordination with all of the other units in their organisation [Ford 1992]. Motivation theory also suggests that it is actually very difficult for people to behave against an organisational ethos.

5.1.2 Structure

A company's approach to quality also seemed to be directly related to the state of the company generally. Companies that had a chaotic internal structure also had a chaotic approach to quality; companies with streamlined and effective internal structures also had streamlined and effective QA programmes. This is very inkeeping with how the CMM rationalises companies and adds weight to the view that companies should only look to certification after they have sorted out their organisational framework.

5.1.3 Size

Only company B was against formal QA: all of the other companies were implementing formal QA to varying degrees with varying levels of success. Company B's stance was partly a function of being very small (and partly because of the maverick approach to formalism taken by the managing director). With only 24 practitioners the managing director can probably keep a reasonable grip on quality without any formality. Whether their approach can sustain quality as the company continues to grow is doubtful. The experience of other companies (for example, Triad Special Systems [Fulton & Myers 1995), suggests that as the company grows a more formal approach to quality will be required.

5.1.4 Commitment

Practitioners all agreed that quality will only be significantly improved by radical culture change. The companies in the study were approaching culture change with varying degrees of commitment:

- company A was succeeding;
- company C was trying hard;
- company D had not really realised that it was necessary;
- company B was determined not to do it.

Again commitment to culture change seemed determined by senior management. Companies with senior managers who wanted a culture change were changing and those which did not were not.

5.1.5 Professionalism

An issue that managers consistently raised was the impact on quality of professionalism. Managers in certified companies said that professionalism was important to quality, while managers in the companies with the least formal QA said they relied exclusively on staff professionalism for quality.

It is, however, of concern that some managers seem to be trying to put responsibly for quality solely on the individual practitioner, while at the same time putting practitioners under intense pressure to deliver software on time and to budget.

5.1.6 Application area

I was surprised that my results do not show a relationship between approaches to quality and application area or whether companies produce software for internal or external customers. The determinants of an effective QA programme seem complex.

5.2 People factors

5.2.1 Previous experience

Previous quality experience had the biggest impact on practitioner's attitudes to quality. There was a strong correlation between practitioners with experience of QA programmes and practitioners who were positive about QA programmes. Ignorance about QA programmes made practitioners anticipate the worst. This was compounded by practitioners feeling that there is not enough accurate, detailed information and guidance on quality. This left them even more ignorant and anxious about the unknown realities of quality programmes. My findings are confirmed by other studies which show that many things never reach practitioners [Hatton 1995] [Pressman 1996]. Practitioners who lacked knowledge about QA programmes worried predominately about bureaucracy and red-tape. They were not anxious about the curtailment of creativity.

Several practitioners in companies with an immature approach to quality had moved from companies with a lot more formalism. These practitioners were, on the whole, advocates of formal QA. They found that developing software was easier with QA.

5.2.2 Technical background

During the study I also got the feeling that trained software engineers were more positive and knowledgeable about QA. Furthermore, practitioners with a computing rather than science background and a software rather than hardware background, were generally most positive about QA.

5.2.3 Management approach

Most middle managers were aware of the dangers of implementing QA initiatives too quickly and too severely. Some were so sensitive about this that they appeared to be using it as an excuse for not tackling quality at all (indeed the organisational theory literature reports this as a common problem). Furthermore, everyone seemed to blame someone else for their company's quality problems, though there was an overriding feeling that: developers wanted improvement to quality working practices; senior managers were not really helpful to quality improvements and middle managers were in the middle trying to satisfy everyone. However, it is interesting to note that although my results portray senior management negatively, they were the only group not represented within the study. Had I predicted how important senior managers were to quality I would have also targeted them. Clearly this is an area of the research that needs following up.

5.2.3 Control

It was also interesting that the developers in company D1 (who had some of their software supplied via an outsourcing contract) felt they had no control over software quality. This is probably a common feeling amongst ordinary buyers of software, and the reason why procurers are so keen on certification. Perhaps if quality software was delivered then the industry would not be under the pressure that it is to certify.

5.3 Resource factors

A significant issue for developers was pressure on quality from unrealistic project planning. Crosby's 'Quality is free' argument [Crosby 1979] had not really penetrated the companies in this study. Deadlines were perceived as much more important to senior managers than quality. It was also clear that managers implicitly condoned evasion of the QA programme when the pressure was on. Indeed only in company A were managers unaware that the QA programme was evaded at times of high pressure (developers said it was). Indeed, several practitioners said that some QA procedures were too easy to avoid and that they would prefer that not to be the case. Again this is similar to what [Neilson & Timmins 1997] found: in one company they observed, one senior manager told staff that quality would have to be put on hold until other 'more pressing' problems were sorted out.

6. Conclusions

6.1 General findings

My results show that software practitioners are keen to see quality improved and to explore ways in which this can be achieved. Developers in almost all of the companies in the study said they wanted a more formal approach to QA. Furthermore, in the companies with least formal QA developers were most keen for more (excluding company B). Although the myth persists that most practitioners are negative about software quality the positive practitioner attitudes that I found are already well documented [DeMarco 1982], [Pfleeger 1993], [Grady & Caswell 1987].

My results show that companies have different approaches to quality improvement and unique quality cultures. I can illustrate this by comparing company A and B. On paper both companies look similar, but in reality their quality cultures could not be more different. I show that a company's quality culture is partly moulded by practitioner attitude to quality, but that practitioner attitude is profoundly influenced by senior management's attitude. My results suggest that the factors traditionally believed to be significant have no discernible impact on a company's approach to QA. For example, whether software is developed for internal or external customers does not seem relevant to a company's approach to QA.

I also de-bunk the myth that practitioners are worried that QA impinges on creativity. I found only a few practitioners who thought that this was an issue.

The experiences, views and concerns of developers and managers were also not clear cut. Managers and developers had different primary QA concerns, but those concerns did not conflict. Both groups were also aware of, and sensitive to, each others concerns. Developers were sympathetic to the pressures that middle managers were under and middle managers were actively trying to address the issues important to developers. Senior managers were the ones who came in for most criticism. Developers and middle managers believed that senior managers were out of touch.

6.2 Evidence in support of the study hypotheses

6.2.1 Hypotheses about the company

i. *QA and measurement are most successful when a quality culture exists within the company.*

I found some evidence in support of this hypothesis. Company A and C had good quality cultures (as described throughout Section Two) and these two companies seemed to have the most effective QA programmes. Company B had no quality culture and had no QA programme.

ii. *It is common for companies not to practice what they claim to in terms of QA and measurement.*

I found some circumstantial evidence in support of this hypothesis. Practitioners in most of the companies in this part of the study said that the QA programme was regularly disregarded when projects were under pressure.

iii. *Companies which have highly structured and disciplined existing working practices are more likely to have effective QA and measurement.*

I found evidence to support this hypothesis in this part of the study. For example, company A had a very structured and disciplined process and also had an effective QA programme. On the other hand, company B had a very undisciplined process and did not have an obviously effective approach to quality.

6.2.2 Hypotheses about the practitioners

i. *Practitioners' attitudes affect the success of QA and measurement.*

I found some evidence of this in this part of the study. Practitioners were most enthusiastic about QA in the companies with the most effective QA programmes (companies A and C). In the company with the weakest QA programme practitioner attitudes towards QA were very negative (company B). However, again, more research is required to identify cause and affect. Are negative perceptions about QA as a result of the company's poor approach to QA? Or are the negative perceptions negatively affecting the effectiveness of the company's QA programme?

ii. *Managers and developers have different views on which QA and measurement practices are worthwhile.*

I found some evidence of this. Although most QA issues were of equal concern to managers and developers across all companies in this part of the study, there were some discrepancies. For example, it was generally felt that mangers were much more interested in quality certification than developers were.

iii. QA and measurement does not allow practitioners to be creative.

I found very little evidence of this. Only practitioners in one company (B) mentioned this as an issue at all.

6.2.3 Hypotheses about the implementation process

i. QA and measurement initiatives are not usually properly resourced.

I found some evidence of this. Developers and mangers in most of the companies in this part of the study (with the notable exception of company A) were concerned that senior managers only paid lip service to QA and were not prepared to commit resources (see section 2.1). This result needs more research.

ii. QA standards are only marketing aids which do not necessarily improve software quality.

Not enough companies were quality certified to examine this hypothesis thoroughly. However, company A had TickIt certification and did not seem to be using it only as a marketing aid. On the other hand company D had ISO9001 certification and did seem to be using it as a marketing aid. More research is needed to follow-up the very mixed views regarding the affect on quality of certification.

Chapter Ten

Summary recommendations for successful quality assurance and measurement

Overall my results suggest that companies implementing effective QA and measurement must develop a *quality culture*. Because this is a rather imprecise and vague term, I present those aspects of implementing QA and measurement that were most effective in the companies in this study, and which most contributed to generating a *quality culture.*

1. Recommendations to companies

The success of all the advice to companies contained in this section is dependent exclusively on effective management. High quality management is the single most important factor to QA and measurement success. Without good management none of this advice can be implemented effectively. This means that companies must, as a priority, have skilful, knowledgeable and well trained managers.

1.1 Establishing management commitment

Managers must really believe in quality, enthusiastically lead the quality initiative and properly resource it. Real, tangible management commitment is an absolutely essential pre-requisite to effective QA and measurement. Practitioners in companies with ineffective QA and measurement often believe that this is senior management's latest bandwagon to which yet more *lip service* will be paid. This approach to quality reveals basic poor management.

1.2 Encouraging practitioner participation

Despite strong evidence to suggest that grassroots initiatives have the most to offer and are the ones that succeed in the long-run, many developers and middle managers feel powerless to effect improvement. Furthermore, many practitioners feel that those quality initiatives imposed by senior management are usually mis-conceived and the result of jumping on the next bandwagon.

This means that companies should:

- Solicit and listen to the QA and measurement views, ideas and experiences of practitioners at all levels in the company.

- Encourage practitioners at all levels in the company to participate in the design and implementation of QA and measurement.

- Implement awareness raising training, as well as detailed technical training for managers and developers.

- Regularly review, assess, evaluate and improve the approaches taken within the company to QA and measurement.

1.3 Slow incremental implementation

The most successful quality initiatives seem to start small and unobtrusively. The most successful programme that I observed was very small and highly focused. The ambitiousness of a programme must also be closely coupled to the maturity of the existing development process. However, it also means that from the outset programmes must have obvious advantages for everyone. Developers and managers must be able to see that they personally will benefit from what is implemented, if this happens then overt and covert resistance from developers and middle managers can be avoided.

On the other hand it is also important to set and control expectations, as practitioners who expect a great deal of benefit very quickly will be disappointed. They may then be turned off the programme altogether.

Piloting is widely believed to be an effective way of implementing QA and measurement.

1.4 Implementing a transparent process

How the programme is operated must be very clear to practitioners at all levels of the company. Furthermore there must be no secrecy or damaging practices within programmes (for example, there must be no link between the programme and staff appraisal). This means it is crucial that effective communication and feedback channels are in place. This kind of transparency will also avoid any discrepancy between 'official' and actual QA and measurement, as what is actually happening should be very obvious.

1.5 Using a goal orientated approach

Companies must be clear about the purpose of QA and measurement, and the operation of the programme must be orientated around that purpose. Furthermore optimal programmes will achieve company and practitioner goal alignment. Within this goal orientated data must be collected. Data collected which is perceived to be inappropriate, inaccurate or being fiddled will seriously damage a programme.

1.6 Limiting pressure on practitioners

Some practitioners seem to be in a no-win quality situation. Quality is being made their professional responsibility at the same time that they are under tremendous pressure to deliver to schedule. Unreasonable project pressure is an anathema to delivering quality. Indeed getting basic company and project management right is a pre-requisite to optimising a programme.

1.7 Understanding the problems of software development

Many practitioners feel that senior management's failure to understand software and software development makes developing quality software that much more difficult.

This means that companies should:

- Adopt a project-based approach to the design and implementation of QA and measurement. This could be based around any software development lifecycle.

- Carry out thorough background research on QA and measurement so as to avoid repeating other companies mistakes and re-inventing the wheel.

2. Recommendations to researchers

2.1 A flexible approach

The most successful quality initiatives not only have flexibility within them, but also have different approaches for different types of project. This flexibility is very popular with practitioners and should be built into research work on QA and measurement programmes.

2.2 Publishing guidance

Most practitioners complain about the lack of really useful QA and measurement guidance. Because of this many practitioners lack knowledge about QA and measurement. This lack of knowledge encourages practitioners to be pessimistic. It also means that companies undergoing certification are unsure about what they really have to implement to gain certification. Indeed the problem of companies taking quality formalism too far could be corrected by more support and guidance from quality bodies.

It is possible that practitioners are simply unaware of the existing literature, but even this situation must be tackled. It may be that access to information is the real problem. However, ignorance is the single most damaging thing to practitioner attitude to QA and measurement.

2.3 Disseminating positive quality experiences

Many practitioners know apocryphal horror stories about QA and measurement. These rumours and exaggerations encourage negative attitudes in practitioners who have no first hand experience of QA and measurement. Very few positive stories of QA and measurement are getting to the practitioners on the ground. This is important as positive experiences play a big role in making practitioners enthusiastic about quality initiatives.

2.4 Discouraging the use of quality certification solely for marketing

Our results suggest that quality standards like ISO9001 are being undermined by companies gaining the certificate only for marketing reasons. Furthermore, many practitioners think that marketing is the major motivation for companies gaining the standard and that quality is not actually improved by the standard.

Chapter Eleven

Conclusions

1. Overall aims

In this thesis I have successfully fulfilled the following aims:

1. Identified best QA and measurement practice in the software industry, as reported in the literature. In doing so I have presented the current software QA and measurement state-of-the art.

2. Reported normal QA and measurement practice in the software industry, as observed in the twenty-six companies that took part in my study and as reported in the literature. In doing this I measured the penetration of the state-of-the-art practices in ordinary companies. Thus I have identified the current software QA and measurement state-of-the-practice

3. Produced guidelines for successfully implementing QA and measurement by analysing and comparing best and normal QA and measurement practice in the software industry. In this analysis I assessed the effectiveness of what is practised.

2. Research hypotheses

Table 11.1 summarises my findings regarding the initial study hypotheses. These findings and their implications are discussed more generally in the next section.

Table 11.1: Testing the study hypotheses: a summary of evidence

	Study hypotheses	Chapter hypothesis tested in	Evidence level found
1	**Hypotheses about the company**		
i	QA and measurement are most successful when a quality culture exists within the company.	8, 9	E
ii	QA and measurement is more likely to be successful where morale in a company is already quite high and practitioners generally feel positive about the company.	6	E*
iii	It is common for companies not to practice what they claim to in terms of QA and measurement.	7, 8, 9	S
iv	Companies which have highly structured and disciplined existing working practices are more likely to have effective QA and measurement.	9	E
v	QA and measurement that is implemented ineffectively will have widespread counterproductive repercussions throughout the company.	6	L
2	**Hypotheses about the practitioners**		
i	Practitioners' attitudes affect the success of QA and measurement.	8, 9	E*
ii	Managers and developers have different views on which QA and measurement practices are worthwhile.	8, 9	E
iii	Managers are enthusiastic about QA and measurement.	6, 8	E
iv	Developers resist the implementation of QA and measurement.	6, 8	L
v	QA and measurement does not allow practitioners to be creative.	9	N
vi	Younger practitioners will be keener on QA and measurement.	6	N
	Highly qualified will be keener on QA and measurement.	6	E
vii	Practitioners with particular technical skills will be more positive about QA and measurement than others. For example, analysts will be more positive than programmers.	6	N
3	**Hypotheses about the implementation process**		
i	QA and measurement initiatives are not usually properly resourced.	9	L
ii	The goals of QA and measurement must be clear to practitioners if the programme is to succeed.	8	N
iii	Effective QA and measurement must be transparent to practitioners if the programme is to succeed.	8	E
iv	The usefulness of QA and measurement should be obvious to all practitioners if the programme is to succeed.	8	E*
v	Developers should participate in designing QA and measurement if the programme is to succeed.	8	E*
vi	QA and measurement initiatives are likely to be most successful when implemented in a bottom-up, incremental way.	-	NP
vii	Practitioners should have confidence in the integrity of measurement data if the programme is to succeed.	8	E
viii	Feedback on QA and measurement needs to be provided if the programme is to be successful.	8	L
ix	Any perceptions that data is being used inappropriately will make practitioners negative about measurement.	8	S
x	QA standards are only marketing aids which do not necessarily improve software quality.	-	NP

Key

N	No evidence found
L	Low evidence
E	Some evidence
E*	Some evidence but the causal relationship needs more research
S	Strong evidence
NP	No data collected

3. Main findings

The main findings of this work include identifying the following:

- The best and normal QA and measurement practice gap.
- The official and actual practice gap.
- The manager and developer QA and measurement perception gap.
- QA and measurement can be managed for success.
- QA and measurement implementation strategies are crucial to success.

I discuss each of these findings in more detail below.

3.1 The best and normal QA and measurement practice gap

There is a significant gap between the QA and measurement practices that companies are recommended to implement, and what is actually practised by most companies. Although a set of best practices for QA and measurement can be identified, best practice (or indeed good practice) is not commonly practised by companies. In this work I have presented evidence about existing baselines for QA and measurement practices and some of my findings are discouraging. The penetration of good practice is generally poor and even procedures that are widely believed to have a significant cost benefit, such as reviews and inspections, have a low take-up.

3.2 The official and actual practice gap

The quality process actually practised in companies is often not what companies claim. Company managers often do not know what is actually in practice. Indeed companies often led me to believe that their programmes were more mature than they were.

3.3 The manager and developer QA and measurement perception gap

There is a gap between groups of practitioners regarding what constitutes effective QA and measurement. Furthermore, I also found evidence to suggest that this gap was often the source of the official and actual practice gap, in that managers believed that a particular process was in place but developers were actually using a different process. I also found that in defining the official QA and measurement programme the views and experiences of developers were not often sought nor listened. Where they were, more effective QA and measurement programmes were usually developed.

3.4 QA and measurement can be managed for success

Throughout this thesis I have shown that practitioners' attitudes to QA and measurement are significant to the ultimate success of a company's programme. Furthermore, these attitudes are dependent on how well managed and effectively implemented programmes are. This means that the attitudes of practitioners towards QA and measurement can be managed and controlled. Thus QA and measurement success can be controlled and managed.

Managers regularly assume that developers will be negative about QA and measurement *per se*, but my results suggest that developers base their views on their QA and measurement experiences. Indeed, as the following quote suggests, I found that the crux of an effective programme was good basic management:

> *"It's a matter of good management of change, if you don't have that you may as well change management!"* [Shelley 1993]

Unfortunately, I have found many instances of poor management in this study. Poor management seems very common and contributes fundamentally to a company's success.

3.5 Implementation strategies are crucial to success

Companies which adopt a well planned and carefully researched implementation strategy are most likely to be successful with QA and measurement. Poor (or non-existent) strategies resulted in programmes that were likely to fail or be ineffective. Furthermore, I also show that effective implementation strategies do exist.

4. The main problems during the research

4.1 Difficulty in obtaining valid data

Not only did it prove difficult to get companies to collaborate in this study, but once companies were collaborating accessing valid data from companies continued to be a problem. Companies were reluctant to participate, probably because they were worried about how they would perform in the study, and they were reluctant to reveal data they considered commercially sensitive.

4.2 A 'prestige bias'

It is probably the case that there are two biases in the data I present, both of which stem from the so-called *prestige bias* [Oppenheim 1992]. The first bias is in the set of companies participating in the study. They were self selecting and consequently companies performing relatively well in terms of QA and measurement were more likely to self select than companies who believed they would perform poorly in this study. The second bias is in the data collected within the companies in the study. Practitioners (managers in particular) tended to exaggerate what the company was doing in terms of QA and measurement. Overall companies seemed very concerned that they should come out well from the study.

4.3 Measuring quality

Unlike previous studies (for example, [Fenton *et al* 1994]), this study does not attempt to quantify the quality of software to relate process quality to product quality. Rather, I tried to establish whether practitioners believed that they produced better software using QA and measurement than not using them. Although such an approach cannot produce convincing scientific evidence, I believe it sheds more light than some other quantitative alternatives, for example, measuring software defect densities.

5. Final comments

My results show that the experiences and views of grassroots practitioners have, until now, not often been sought or listened to. I show that although in the last 5-10 years there has been significant theoretical progress in the field of software quality, this has not always penetrated into ordinary companies. Indeed, I found that many practitioners are left frustrated by poor quality but lack the power and information to do much about it. This is despite practitioners having clear ideas about how things could be improved. The real experiences of many practitioners is that the market wants cheap software as quickly as possible and is not too concerned about the quality of it. Furthermore the current commercial climate makes it difficult for companies to stop 'firefighting' and plan for long term quality improvement.

Overall, the ineffective way that even some of the best companies have gone about implementing QA and measurement suggests that the industry still has some way to go before it really does mature into a well managed industry which produces high quality products. However, some of the experiences described in this work do show that, if the introduction of QA and measurement into a software development environment is sensitively managed, then real dividends can be expected certainly in the medium term and probably in the long term.

6. The future of this study

Overall I have been successful in this research and have been awarded a three year EPSRC grant of £143k to fund the further development of the work described in this thesis. Consequently I will be able to follow-up many of the questions this research has generated.

References

Albrecht A, Gaffney J (1983) “Software Function, Sources Lines of Code and Development Effort Predication” IEEE Trans on Software Engineering, 9(6) pp639-648

Anderson A, Kypnanou A (1994) “Effective Organisational Behaviour” Blackwell

Bache R, Neil M (1995) "Introducing Metrics into Industry: A Perspective on GQM" in [Fenton, Whitty & Iizuka 1995] pp59-67

Bacon S (1994) “ami at Provident Mutual” in Procs of ami User Group, 24th March, Edinburgh

Baker MD (1991) “Implementing an Initial Software Metrics Programme” National Aerospace and Electronics Conf. IEEE Procs, vol 3, pp1289-1294

Barker H (1995) “The SPICE Technical Architecture” AIC Conference, London March 13-14

Basili VR (1995) “Applying the GQM Paradigm in the Experience Factory” in [Fenton, Whitty & Iizuka 1995] pp23-39

Baumert JH, McWhinney MS (1992) "Software Measures and the Capability Maturity Model" SEI Technical Report, CMU/SEI-92-TR-25

Bazzana G, Caliman P, Gandini D Giunchi M, Lanellotti R, Marino P (1995) "Software Management by Metrics: Practical Experiences in Italy" in [Fenton, Whitty & Iizuka 1995] pp185-193

Berdie D, Anderson J (1974) “Questionnaires: Design and Use” Scarecrow Press

Billings C, Clifton J, Kolkhorst B, Lee E, Wingert WB (1994) “Journey to a Mature Software Process” IBM Systems Journal, vol 33(1), pp46-61

Boehm B (1981) “Software Engineering Economics” Prentice Hall, New York

Bologna S, Bucci G (1996) "Achieving Quality in Software"
Chapman & Hall

Bray M, Ross M, Staples G (eds) (1996) "Improving Quality"
Mechanical Engineering Publications

Bryman A, Cramer D (1994) "Quantitative Analysis for Social Scientists"
Routledge

Bryman A (1990) "Research Methods and Organisational Studies"
Unwin Hyman

Burgess (1991) "Field Research: A Sourcebook and Field Manual"
Routledge

Card D (1988) "The Role of Measurement in Software Engineering"
IEE Conf Publication No 290, pp33-37

Carlson W, McNurlin B (1992) "Basic Principles for Measuring IT Value"
I/ S Analyser, vol 30, pp1-14

Constantine LL, Lockwood LAD (1994) "Fitting Practices to the People"
American Programmer, Dec. vol 12, pp21-27

Cougar JD , Zawacki RA (1980) "Motivating and Managing Computer Personnel"
Wiley

Cox GM (1991) "Sustaining a Software Metrics Programme in Industry"
in [Fenton & Littlewood 1991] pp1-15

Crosby P (1979) "Quality is Free"
McGraw Hill

Curtis B, Hefley W, Kondrad M, Miller S (1995) "People Capability Maturity Model"
CMU/SEI-95-HB-02, SEI Carnegie Mellon University

Daskalantonakis MK (1992) "A Practical View of Software Management and Implementation Experiences within Motorola" IEEE Transactions on Software Engineering, 18 (11), pp998-1009

Davidson C (1991) "Measuring Up"
Computer Weekly, 7th Nov

Dell T (1989) "How to Motivate People: Principles and Applications"
Kogan Page

DeMarco T (1995) "Why does Software cost so much?"
Dorset House Publishing

DeMarco T (1982) "Controlling Software Projects"
Yourdon Press, Prentice Hall

Dion R (1992) "Elements of a Process Improvement Programme"
IEEE Software, Quality Time Column, July, pp83-85

Drummond-Tylor E (1989) "Software Metrics: An International Survey of Industrial Practice"
ESPRIT 2 Project METKIT Sema Group

Dti (1992) "Guide to Software Quality Management System Construction and Certification using EN29001", DISC TickIT Office

Easterby-Smith M, Thorpe R, Lowe A (1991) "Management Research - An Introduction"
Sage Publications

Espley R (1993) "Industrial Experience - Working with AMI"
Procs of 10th Annual CSR Workshop, Amsterdam

Fagan ME (1976) "Design and Code Inspections to Reduce Errors in Program Development"
IEEE TSE 1(2), pp140-149

Fenick S (1990) "Implementing Management Metrics: An Army Program"
IEEE Software, March vol 7(2), pp65-72

Fenton N (1996) "Do Standards Improve Quality?"
Counterpoint Column, IEEE Software, Jan 13(1) pp22-24

Fenton N, Pfleeger SL (1996) "Software Metrics: A Rigorous and Practical Approach"
(2nd Edition), International Thomson Computer Press

Fenton N, Whitty RT, Iizuka I (eds) (1995) "Software Quality Assurance and Metrics. A Worldwide Perspective" Thompson Computer Press

Fenton N, Pfleeger SL, Glass R (1994) "Science and Substance: A Challenge to Software Engineers" IEEE Software, July, pp86-95

Fenton N, Littlewood B (1991) "Software Reliability and Metrics" Elsevier Applied Science

Fenton N (1991) "Software Metrics: A Rigorous Approach" Chapman & Hall

Flowers S (1996) "Software Failure: Management Failure" Wiley

Ford ME (1992) "Motivating Humans" Sage Publications

Fransella F, Bannister D (1977) "A Manual for Repertory Grid Technique" Academic Press

Frude N (1993) "A Guide to SPSS/PC+" McMillan

Fuchs N (1995) "Software Measurement: An Evolutionary Approach" in [Fenton, Whitty & Iizuka 1995] pp69-77

Fulton AM, Myers BM (1995) "Should 'Quality' be Dropped from 'Quality Management Systems'?" in [Ross *et al* 1995] pp15-26

Gilb T, Graham D (1994) "Software Inspections" Addison Wesley

Gillies (1992) "Software Quality: Theory and Management" Chapman & Hall, London

Glass RL (1994) "From Wonderland to the Real Problem" IEEE Software, Manager Column, May pp190-193

Goodman P (1995) "The Practical Implementation of Process Improvement Initiatives" in [Fenton, Whitty & Iizuka 1995] pp148-156

Goodman P (1992) "Implementing Software Metrics Programmes: A Project Based Approach" Eurometrics '92 Int Conf. pp 147-155

Gouldner AW (1954) "Patterns of Industrial Bureaucracy" Free Press, New York

Grady RB (1992) "Practical Software Metrics for Project Management and Process Improvement" Prentice-Hall

Grady RB, Caswell DL (1987) "Software Metrics Establishing a Company-wide Program" Prentice Hall

Hambling B (1996) "Managing Software Quality" McGraw Hill

Handy C (1993) "Understanding Organisations" Penguin

Hatton L (1995) "Automated Incremental Improvement of Software Product Quality: A Case History" in [Fenton, Whitty & Iizuka 1995] pp228-241

Hawkins C, Ross M, Staples G (1997) "The Quality Challenge" Mechanical Engineering Publications

Hetzel B (1995) "The Sorry State of Software Practice, Measurement and Evaluation" in [Fenton, Whitty & Iizuka 1995] pp97-108

Hetzel B (1993) "Making Software Measurement Work - Engineering an Effective Measurement Program" QED

Humphrey WS, Snyder TR, Willis RR (1991) "Software Process Improvement at Hughes Aircraft" IEEE Software, vol 8(4), pp11-23

Humphrey WS (1989) "Managing the Software Process" Addison Wesley, Reading, Mass.

IEEE (1994) Proceedings of the Second International Software Metrics Symposium
IEEE Computer Society Press

IEEE (1983) Standard 729: Glossary of Software Engineering Terminology
IEEE Computer Society Press

Iizuka Y (1995) "A New Paradigm for Software Quality: The Turning Point for the Japanese Software Industry" in [Fenton, Whitty & Iizuka 1995] pp110-119

ISO (1991a) "ISO 9000-3 Quality Systems. Guide to the Application of ISO 9001 to the Development, Supply and Maintenance of Software"

ISO (1991b) "ISO 9899. The C Programming Language"

Jeffery R, Berry M (1993) "A Framework for Evaluation and Prediction of Metrics Program Success" 1st Intl Software Metrics Symp, Baltimore MD, IEEE Computer Society Press, pp28-39

Jones C (1996) "Our Worst Current Development Practices"
IEEE Software, Manager Column, pp102-104

Jones C (1994) "Software Management: The Weakest Link in the Software Engineering Chain"
IEEE Computer, May, pp10-11

Jones C (1991) "Applied Software Measurement"
McGraw Hill

Juran JM, Gryna FM Jr, Bingham FM (eds) (1979) "Quality Control Handbook"
McGraw Hill, New York, 3rd Edition

Kanter RM (1984) "Change Master - Corporate Entrepreneurs at Work"
George, Allen & Unwin

Karalak D (1993) "Metrics Based Software Engineering at ITT Aerospace Communications"
Applications of Software Metrics, 4th Int. Conf. Florida, pp184-198

Kearney AT (1992) "Total Quality: Time to take off the rose tinted spectacles"
The TQM Magazine, Feb

Keinayel L, Mili A, Ouederni I (1991) "Controllable Factors for Programmer Productivity: A Statistical Study" Journal of Systems Software, vol 16, pp151-163

Kettler E (1993) "Implementing Metrics in a Large Organisation" Applications of Software Metrics, 4th Int. Conf. Florida, pp560-575

Kelly (1955) "The Psychology of Personal Constructs" Norton, New York

Kitchenham B, Pickard L, Lawrence Pfleeger S (1995) "Case Studies for Method and Tool Evaluation" IEEE Software, 12(4), pp52-63

Kitchenham B (1996) "Software Metrics: Measurement for Software Process Improvement" NCC Blackwell

Kitchenham B, McDermid JA (1986) "Software Metrics and Integrated Support Environments" Software Engineering Journal, Jan, pp58-64

Kitchenham B (1984) "Program History Records: A System of Software Data Collection and Analysis" ICL Technical Journal, May, pp103-114

Konrad MD, Paulk M, Graydon A (1995) "An Overview of SPICE's Model for Process Management" in Procs 5th Intl Conf on Software Quality, Oct, Austin Texas, pp291-303

Lindstrom DR (1993) "Five Ways to Destroy a Development Project" IEEE Software, Sept, pp55-58

Lytz R (1995) "Software Metrics for the Boeing 777: A Case Study" Software Quality Journal, 4(1), pp1-14

McCrossan L (1985) "A Handbook for Interviewers" HMSO

McGarry J (1996) "Practical Software Measurement: A Guide to Objective Program Insight" Joint Logistics Commanders Joint Group on Systems Engineering

McNeil P (1990) "Research Methods" Routledge

Marsh C (1982) "The Survey Method"
Allen and Unwin

Martin J, Myerson DE (1988) "Organisational Culture and the Denial, Channelling and Acknowledgement of Ambiguity" in [Moch *et al* 1988]

Moch M *et al* (eds) (1988) "Managing Ambiguity and Change"
Wiley, New York

Möller K (1993) "Software Quality Metrics Application"
Applications of Software Metrics, 4th Int. Conf. Florida, pp726-748

Myerson DE, Martin J (1987) "Culture Change: An Integration of three Different Views"
Jnl of Management Studies, vol 24, pp623-647

NCC (1993) "An Introduction to Software Metrics"
NCC Fact Sheet

Neil M (1994) "Measurement as an Alternative to Bureaucracy for the Achievement of Software Quality" Software Quality Journal, 3(2), pp65-78

Neilson S, Timmins R (1997) "A Framework for Analysing Organisational Commitment to Software Quality Management" in [Hawkins *et al* 1997]

Neilson S (1994) "Software Quality Management and Organisational Fit"
in Procs of 5th Australasian Conf. on Information Systems, Melbourne, Sept. pp493-503

Nicholson IA, Homes D, Richards J, Warden R (1995) "Motivation of Quality Practitioners"
in [Ross *et al* 1995] pp349-359

Oppenheim AN (1992) "Questionnaire Design and Attitude Measurement"
Open University Press

Ormrod J (1990) "Time for Metrics"
EXE, Sept vol 5(4), pp48-52

Pfleeger SL (1995) "Setting up a Metrics Programme in Industry"
in [Fenton, Whitty & Iizuka 1995] pp45-59

Pfleeger SL, Fenton NE, Page P (1994) "Evaluating Software Engineering Standards" IEEE Computer, Sept, pp71-79

Pfleeger SL (1993) "Lessons Learned in Building a Corporate Metrics Program" IEEE Software, 10(3), pp67-74

Pfleeger SL, McGowan CL (1990) "Software Metrics in a Process Maturity Framework" Journal of Systems and Software, vol 12, pp255-260

Pike J, Barnes R (1996) "TQM in Action: A Practical Approach to Continuous Performance Improvement" Chapman & Hall

Porter LJ, Parker AJ (1993) "Total Quality Management - The Critical Success Factors" The TQM Magazine 14(1)

Pressman R (1996) "Software Process Impediment" Manager column, IEEE Software, Sept, pp16-18

Pulford K, Kuntzmann-Combelles A, Shirlaw S (1996) "A Quantitative Approach to Software Management: The ami Handbook" Addison Wesley

Rombach DH, Basili VR, Selby RW (1993) "Experimental Software Engineering Issues" Springer Verlag

Rombach DH (1990) "Design Measurement: Some Lessons Learned" IEEE Software, 7(2), pp17-25

Ross M, Brebbia CA, Staples G, Stapleton J (eds) (1995) "Software Quality Management III" Computational Mechanics Publication

Ross M, Brebbia CA, Staples G, Stapleton J (eds) (1993) "Software Quality Management " Computational Mechanics Publication

Rouse A, Watson D (1994) "The Role of Culture in IS Quality" in Procs of 5th Australasian Conf on Information Systems, Melbourne, Sept. pp405-415

Rubin H (1995) "IT Metrics Newsletter" Vol I(4)

Rubin H (1990) "The Rubin Review" Vol III(3)

Russell M (1990) "International Survey of Software Measurement Education and Training" Journal of Systems and Software, vol 12, pp233-41

Shelley CC (1993) "Practical Experiences of Implementing Software Measurement Programmes in Industry" in [Ross *et al* 1993] pp95-107

Sommerville I (1996) "Software Engineering" McGraw Hill, 4th Ed

Stevenson C (1993) "A Poor Man's Metric" Applications of Software Metrics, 4th Int. Conf. Florida, pp200-222

Stockman S (1993) "Experiences in the Introduction of Software Measurement" Procs of 10th Annual CSR Workshop, Amsterdam

Tuttle T, Sink D (1984/85) "Taking the Threat out of Productivity Measures" National Productivity Review, Winter, pp24-32

Valett JD, McGarry FE (1988) "A Summary of Measurement Experiences in the Software Engineering Laboratory" 21st Hawaiian Int Conf on System Science, pp293-312

Vernier J, Moores T, Barrett R (1996) "Software Quality: Perceptions and Practices in Hong Kong" in [Bologna & Bucci 1996] pp77-89

Warner A, Janson J, Piekos J (1995) "The Rise and Fall of the High Performance Team" in Procs of 5th Intl. Conf. on Software Quality, Oct, Austin, Texas, pp217-229

Weber M (1947) "The Theory of Social and Economic Organisation" The Free Press, New York

Weinberg G (1993) "First Order Measurement" Quality Software Management Vol II, Dorset House

Weinberg G, Schulman E (1974) "Goals and Performance in Computer Programming" Human Factors, 16(1), pp70-77

Weller E (1993) "Lesson from Three Years of Inspection Data" IEEE Software, Sept, pp38-45

White L (1993) "The Human Factors of Measurement"
Applications of Software Metrics, 4th Int. Conf. Florida, pp126-137

Wilkinson A, Witcher B (1993) "Holistic Total Quality Management Must take Account of Political Processes" The TQM Magazine 4(1)

Wilson DN, Dawson C (1996) "Process and Product Quality: Perpetuating the Myth"
in [Bray *et al* 1996] pp135-145

Wilson DN, Petoc P, Roiter K (1995) "A Study of Software Quality Assurance in Practice"
in [Ross *et al* 1995] pp159-171

Appendix One

List of companies

Company	Reference
Boeing	[Lytz 1995]
British Telecom (BT)	[Stockman 1993]
Contel	[Pfleeger 1993]
EDS/General Motors	[Kettler 1993]
GEC Marconni	[Espley 1993]
Hewlett Packard (HP)	[Grady & Caswell 1987]
Hughes Aircraft	[Humphrey *et al* 1991]
ICL	[Kitchenham 1993]
Infomix	[Bache & Neil 1994]
ITT Aerospace	[Karalak 1993]
Kodak	[Carlson & McNurlin 1992]
Mitre Corp	[Clapp 1990]
Motorola	[Daskalanakis 1992]
NASA/IBM	[Billings *et al* 1994]
NASA/SEL	[Rombach 1990]
National Computer Systems (NCS)	[Stevenson 1993]
NCC	[NCC 1993]
Provident Mutual	[Bacon 1994]
Paramax Electronic Systems	[Lindstrom 1993]
Racal	[Shelley 1993]
Raytheon	[Dion 1992]
Siemens	[Möller 1993]
Software Engineering Laboratory (SEL)	[Valett & McGarry 1988]
US Army	[Fenick 1990]
US Navy	[McGarry 1996]

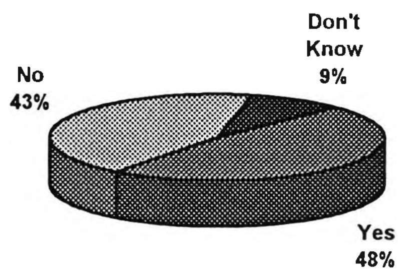
Appendix Two

A pilot company case study

1. Introduction

In this appendix I present the results from the first phase of this study. This phase was designed to test and improve my study methodology. However, it also provides an interesting insight into QA and measurement in the pilot department and roughly indicates which areas of research seem to be worth following up in subsequent phases of the study. I applied the whole research lifecycle during the pilot, and this is fully described in the Study Methods chapter (Chapter Two).

2. Job satisfaction and attitude to measurement



Do you think that the collection of statistical data in software development is useful?

Figure A3.1: Perceived data usefulness

Figure A3.1 shows that there was no consensus amongst practitioners on whether measurement in software development was a useful activity. I did, however, find some promising relationships when I looked at the attitudes practitioners had to their jobs and the particular feelings they had towards QA and measurement. Although twenty practitioners (87%) said that they enjoyed their job, there was a difference in attitude towards QA and measurement between those who did, and those who did not enjoy their job. Two thirds of the practitioners who said that they did not enjoy their job, were negative about the introduction of QA and measurement. In addition, all of the practitioners who did not enjoy their jobs:

- are not always able to do high quality work (the overall average being 68%)
- are affected by motivation problems (the overall average being 46%)
- do not believe that the Department provides a framework for improving working practices (the overall average being 25%)

Features of the eleven practitioners (48%) who were positive about QA and measurement includes them:

- working for the Department between 3-5 years (36% compared to an overall average of 25%)
- ed between 25-34 years (54% compared to an overall average of 33%)
- being male (73% compared to an overall average of 65%)
- being qualified to at least degree level (91% compared to an average of 58%)

Of the practitioners who were negative about QA and measurement, the following comment is typical:

> *"Experience and feeling are more useful "*

Practitioners also had revealing comments to make about using more procedures and standards in software development. Comments included:

> *"I cannot use procedures as I have to do everything as quickly as possible"*
> *"Formal procedures would be too inflexible for me"*

3. Management and measurement

Of the twenty-three practitioners describing themselves as managers, I found some interesting and unexpected results:

- **Managers were less keen on QA formalism than developers.**
  55% of managers thought that formal procedures would be useful in their job, compared to 87% of developers. 54% of managers felt that the quality of their work was never affected by the lack of written procedures compared to 22% of developers.

- **Managers' and developers' attitudes to measurement were similar.**
  I expected to find that managers were keener on measurement than developers, but this was not the case in this Department.

- **Managers were more positive than developers about existing QA procedures.**
  37% of developers thought that a good process improvement framework was present in the Department, compared to 78% of managers. 58% of developers considered that the Department was efficient, compared to 89% of managers.

  In response to the question "*Are you always able to provide a quality service or product?*" although only 22% of all practitioners said that they were *always* able to, 61% of those were managers (despite, 61% of practitioners saying they were, at some time, affected by poor management). Managers also thought that the Department was delivering a higher quality service than developers did. 41% of all practitioners thought that the Department was delivering high quality products and services, 78% of whom were managers. I suspect that this was managers displaying 'prestige bias' as described in [Oppenheim 1992], a situation that came up in every subsequent phase of the study.

From these results I concluded that developers seemed more aware of existing quality problems. Managers seemed to have an exaggerated view of how well the Department was performing. Managers also gave me the impression that they believed that although they personally do not need QA procedures, developers do.

4. The quality dichotomy

Although all the practitioners said they were keen to improve quality, 68% of practitioners said that they were not always able to deliver a high quality service or product. Below is a ranked list of the factors which practitioners believe affected their ability to do high quality work:

1. Lack of information (92%)
2. Lack of training (83%)
3. Too much work (83%)
4. Lack of resources (79%)
5. Lack of working procedures (67%)
6. Lack of written procedures (62%)
7. Poor management (62%)

I found that the most serious impediment to quality improvement, was a perceived lack of information within the Department. Indeed, although 84% of practitioners in the Department said that they are keen to see quality improve, there does seem to be some confusion regarding the QA procedures that are already in place. Figure A3.2 shows that most practitioners do not know whether there are any QA procedures in place or not.

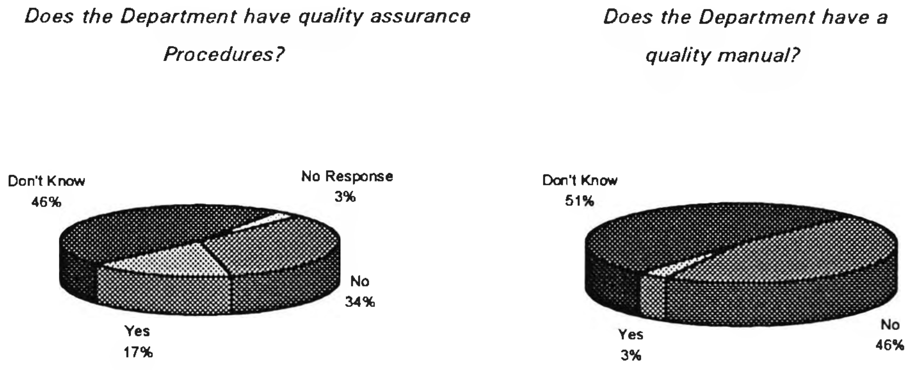


Figure A3.2: Quality assurance procedures

This lack of information is also apparent elsewhere in the pilot study. In response to many questions practitioners have answered *Don't know*, for example 29% of practitioners did not know whether the Department provided good quality products or services.

5. Working practices

I examined a number of possible relationships between working practices and attitude to QA and measurement. I found no significant relationship between practitioners already using formal procedures and practitioners valuing the use of procedures: 43% of practitioners claimed to be working within formal written procedures; 71% of those practitioners thought that such procedures were useful; 78% of practitioners who did not use procedures thought that procedures would be useful. Similarly, 45% of the practitioners who were positive about measurement, are already collecting data on time sheets, but the 55% who did not keep time sheets did not think that measurement is a good idea. These results suggest that some aspects of QA and measurement are very Department specific and dependent on how the Department is organised and how QA has been tackled up until now.

6. Conclusions

There are four conclusions which I have tentatively drawn from this preliminary study and which I plan to follow up in the main study. These are:

1. **Education.** Practitioners' views about QA and measurement seem to be influenced by education. I found that practitioners who were well qualified had more positive attitudes.

2. **Job satisfaction.** This factor seems to affect attitudes to QA and measurement. Practitioners who said that they did not enjoy their job, were also more likely to be negative about QA and measurement.

3. **Management responsibility.** Managers and developers seemed to have different attitudes to QA and measurement. My results suggest that managers:

    - Are less prepared than developers to acknowledge the real state of quality within their Department.
    - Do not think that QA procedures are relevant to them personally.

4. **Assertion.** Claims about quality are made on no quantifiable basis.

Many of the QA and measurement problems I highlight are, probably, common. However, my results do show that practitioners want to improve the work that they are doing, and are interested in ways that this might be achieved.

Appendix Three

Questionnaires

Please tick the answer of your choice. Feel free to attach an additional sheet in order to comment further however.

ALL RESPONSES WILL BE TREATED IN STRICTEST CONFIDENCE.

Please return questionnaires to : TRACY HALL, School of Computing

1. How many years have you worked for Computing Services?

   a) 0-2 ☐ b) 3-5 ☐ c) 6-10 ☐ d) Over ten ☐

2. How old are you?

   a) Under 25 ☐ b) 25-34 ☐ c) 35-44 ☐ d) 45-54 ☐

   e) Over 55 ☐

3. Are you? a) Female ☐ b) Male ☐

4. What is the highest qualification that you have have?

   a) GCE's/GCSE's ☐
   c) 'A' Levels ☐
   d) HND ☐
   e) Degree ☐
   f) Professional qualifications. ☐

   Please specify ....

5. What is your job ?

   a) Analyst/Programmer ☐
   b) Operator ☐
   c) Systems Manager ☐
   d) User Support ☐
   e) Programmer ☐
   f) Analyst ☐
   g) Other. ☐

   Please specify ....

5. Do you supervise others in your job ?

   a) Yes ☐ b) No ☐

6. Do you enjoy your job? a) Yes ☐ b) No ☐

   Please comment on your answer ....

   ....

7. Is your job well defined and structured so that you know what is expected or you?

   a) Yes ☐ b) It's OK ☐ c) No ☐

   Please comment on your answer ....

   ....

8. How frequently do you engage in each of the following as part of your job?

		a) Never	b) Occasionally	c) Frequently
i	Installing and supporting software packages			
ii	User support			
iii	System management			
iv	Systems analysis			
v	System design			
vi	3 or 4GL programming			
vii	System Maintenance			
viii	System testing			

ONLY ANSWER QUESTIONS 9, 10 AND 11 IF YOU ARE OCCASIONALLY, OR FREQUENTLY, INVOLVED IN THE DEVELOPMENT OF SOFTWARE

9. How often to you use any of the following methodologies or practices in your job?

		a) Never	b) Occasionally	c) Frequently
i	CASE tools			
ii	The "Waterfall" Software Lifecyle			
iii	The "V" Software Lifecycle			
iv	SSADM			
v	Structured programming			
vi	Acceptance testing			
vii	Prototyping			

10. How frequently do you use any of the following standards?

		a) Never	b) Occasionally	c) Frequently
i	Documentation standards			
ii	Programming standards			
iii	Test standards			
iv	Configuration Management			
v	Other.			

Please specify ....

11. Which of the following statements do you agree with ?

	Agree	Disagree	Don't know
i. Programming is inherently difficult and is essentially a creative activity			
ii. Adopting a firmly structured and managed approach to programming will lead to good programs being written.			

12. Which of the following statements do you agree with?

	Strongly Agree	Agree	Uncertain	Disagree	Strongly Disagree
i. Computing Services is highly efficient					
ii. Computing Services satisfies its user's requirements					
iii. Computing Services provides high quality products and services					
iv. Computing Services provides the organisational framework for people to function effectively in their job					
v. Most Computing Services staff are keen to see product and service quality continue to improve					
vi. Only the management of Computing Services is keen to see product and service quality continue to improve					
vii. Quality does not seem to be an important issue in Computing Services					
viii. The supportive working environment of Computing Services allows for the continual improvement of working practices					

13. Are you always able to provide a quality service or produce a quality product ?

a) Yes ☐ b) No ☐

14. How do the following factors negatively affect the quality of the work that you are able to do ?

	a) Regularly Affects	b) Occasionally Affects	c) Never Affects
i. Lack of training			
ii. Lack of clear working procedures			
iii. Lack of written procedures			
iv. Lack of information			
v. Too much work			
vi. Lack of motivation			
vii. Lack of resources			
viii. Poor management			
ix. Other.			

Please specify ....

15. Does Computing Services use any of the following?

	a) Yes	b) No	c) Don't know
i. Quality Control procedures			
ii. Quality Assurance procedures			
iii. Quality Manual			
iv. Software Metrics			

16. Are there any formal, written procedures that you follow in executing your work?

a) Yes ☐ b) No ☐

17. Are there any informal, unwritten procedures that you follow in executing your work ?

a) Yes ☐ b) No ☐

Please comment on your answer...

....

18. Do you feel that formal procedures in your job would be, or are, useful ?

a) Yes ☐ b) No ☐

Please comment on your answer ....

....

19. Are you required to record any of the following statistical data ?

	a) Yes	b) No
i. Time Sheets	☐	☐
ii. Programming data (for example the number of lines of code written)	☐	☐
iii. Fault data (for example the number of bugs fixed in a piece of software)	☐	☐
iv. User data (for example the number and class of users for a particular piece of software.	☐	☐

If you record any other statistical data, or have answered "YES" to any of the above, please attach more information, if possible.

20. Do you feel that recording statistical data is a useful activity?

a) Yes ☐ b) No ☐

Please comment on your answer ....

THANK YOU FOR COMPLETING THIS QUESTIONNAIRE.

TH/JK/12/51(6) 39

THE IMPLEMENTATION OF SOFTWARE METRICS

A SURVEY OF CURRENT INDUSTRIAL PRACTICE

2nd International Symposium on Software Metrics: Bridging the Gap Between Research and Practice

Section One: About this Survey

Dear Colleague

This survey forms part of a comprehensive study into the critical success factors of introducing software metrics. The aim of the overall study is to devise a "people-friendly" strategy for the successful introduction of software metrics. The study is looking at both the experiences of organisations who have already introduced metrics, and the differing attitudes developers and managers have to the introduction of software metrics.

Your experiences of software metrics are invaluable to this study so please take the time to complete this questionnaire.

All responses will be treated in total confidence.

Many thanks.

Tracy Hall

Section Two: About Yourself

1. How long have you worked for your current organisation ?

☐ up to one year ☐ one to five years ☐ five to ten years ☐ ten years or more

2. How old are you ?

☐ less than 25 ☐ 25 - 34 ☐ 35 - 44 ☐ 45 - 54 ☐ over 54

3. Are you ? ☐ female ☐ male

4. What is the highest qualification that you have ?

☐ none ☐ High School ☐ Degree ☐ Higher Degree

5. Which one of the following best describes your current job ?

☐ Analyst ☐ Programmer ☐ Analyst/Programmer ☐ Tester
☐ Senior Analyst ☐ Senior Programmer ☐ Manager ☐ Quality Assurer/Controller/Manager
☐ Researcher ☐ Lecturer/academic
☐ Other Please specify ______________________________

Section Three : About Your Organisation

In this section your 'organisation' refers to the company or the institution that you work for directly.

6. How many people does your whole organisation employ ?

☐ under 100 ☐ 100 - 500 ☐ 500 - 1000 ☐ over 1000

7. How many of your organisation's employees jobs are to do with software ?

☐ none ☐ under 10 ☐ 10 - 100 ☐ 100 - 500 ☐ over 500

8. **Tick as many of the following characteristics** that apply to the kind of software development work that your organisation is **primarily** involved in :

☐ in-house developments
☐ data Processing work
☐ finance
☐ software house
☐ safety critical work
☐ research and development
☐ public sector organisation
☐ consultancy

If you do not feel that the above categorisation adequately reflects your organisation then please give more detail below :

____
____
____

9. In your opinion, which of the following best describes your organisation's approach to quality ?

☐ very committed, provides the resources, staff and structures for quality to be constantly improved
☐ committed, but does not have the resources to improve quality as much as it could
☐ only pays only lip service to quality
☐ is not really interested in quality
☐ other, please specify ____

Section Four: About Your Department

In this section your 'department' refers to the department, section or team that you work within. In most cases this will mean the smallest operational work group to which you belong.

10. How often does your department use the following standards ?

	Never	Occasionally	Frequently
Documentation standards	☐	☐	☐
Programming standards	☐	☐	☐
Test standards	☐	☐	☐

16. How frequently does your Department use the following :

	Never	Occasionally	Frequently
Specification Inspections or Reviews	☐	☐	☐
Design Inspections or Reviews	☐	☐	☐
Code Inspections or Reviews	☐	☐	☐
Document Inspections or Reviews	☐	☐	☐
Quality Manual	☐	☐	☐

17. Would you say that your Department has any of the following approaches to software development ?

	Yes	No	Don't know
Highly structured	☐	☐	☐
Highly managed	☐	☐	☐
Creativity-driven	☐	☐	☐
Quality-focused	☐	☐	☐

18. Do you think that your Department produces high quality software ?

☐ always ☐ usually ☐ sometimes ☐ rarely ☐ never

19. Are the overall goals of your Department made clear to you ?

☐ always ☐ usually ☐ sometimes ☐ rarely ☐ never

20. In general, how high is staff morale in your Department ?

☐ very high ☐ high ☐ neither high nor low ☐ low ☐ very low

21. Does your Department have BS5750/ISO9000 certification ?

☐ yes ☐ no ☐ don't know

22. What level of maturity, according to the SEI 'Capability Maturity Model', has your Department reached ?

☐ level one ☐ level two ☐ level three ☐ level four ☐ level five ☐ don't know

Section Five : About Software Metrics

23. Which of the following best describes your organisation's current position to software metrics ?

☐ no metrics are collected ☐ random/sporadic collection ☐ widespread use in your Department
☐ widespread Organisation-wide programme ☐ metrics collection has now been abandoned

If you answered **'no metrics are collected'** to the last question then :

a) Please give a brief indication as to why **you think** that metrics have not been collected so far :

____

____

____

____

b) Go to Section Seven of this questionnaire

24. How long has your Department been collecting software metrics ?

☐ less than one year ☐ one to five years ☐ five to ten years ☐ ten years or more ☐ don't know

25. How are you personally involved in metrics ? (tick as many as apply)

☐ collection ☐ data analysis ☐ management ☐ definition ☐ not involved

26. Does your organisation record any of the following metrics information ?

Analysis Metrics :	Yes	No	Don't Know
Function points	☐	☐	☐
Size data	☐	☐	☐
Cost data	☐	☐	☐
Effort data	☐	☐	☐
Fault data	☐	☐	☐
Design Metrics :			
Effort data	☐	☐	☐
Fault data	☐	☐	☐
Code Metrics :			
Module Interface data	☐	☐	☐
Complexity measures	☐	☐	☐
Lines of Code data	☐	☐	☐
Data-flow data	☐	☐	☐
Other Module data	☐	☐	☐
Effort data	☐	☐	☐
Fault data	☐	☐	☐
Fault Metrics :			
Fault rates	☐	☐	☐
Defect densities	☐	☐	☐
Change data	☐	☐	☐
Change effort data	☐	☐	☐
Inspection data	☐	☐	☐

If any other information is collected in your organisation then please give details on the last page of this questionnaire.

27. Does your organisation have ...

	Yes	No	Don't Know
an organisational metrics database ?	☐	☐	☐
a departmental metrics database ?	☐	☐	☐
automated metrics collection ?	☐	☐	☐
a metrics co-ordinator, manager or promoter ?	☐	☐	☐

28. Has your organisation used 'AMI' ? ☐ yes ☐ no ☐ don't know

29. How much time, on average, do <u>you</u> spend recording metrics data during your working week ?

☐ no time ☐ less than 1 hour ☐ 1 - 5 hours ☐ 6 - 10 hours ☐ over 10 hours

30. Does your organisation provide feedback to the people who collect metrics data ? ☐ yes ☐ no ☐ don't know

<u>Section Six : What is Your Opinion of the way Software Metrics are used in your Organisation ?</u>

31. How carefully is software metrics data collected in your organisation to ensure accuracy ?

☐ very accurately ☐ quite accurately ☐ quite inaccurately ☐ very inaccurately ☐ don't know

32. Does your organisation use the metrics data that has been collected ...

	Yes	No	Don't Know
to assess peoples' individual performance	☐	☐	☐
to assess the organisation's performance	☐	☐	☐
to determine peoples' pay	☐	☐	☐
to assess the quality of software	☐	☐	☐
to assess the software development process	☐	☐	☐
as a project management tool	☐	☐	☐

If your organisation uses metrics data for other things please give some details on the last page of this questionnaire.

33. How much of the metrics data that your organisation collects do you think is used ?

☐ none ☐ less than half ☐ most ☐ all ☐ don't know

34. How much of the metrics data that your organisation collects do you think is the right data to collect ?

☐ none ☐ less than half ☐ most ☐ all ☐ don't know

35. Do you think that there is any manipulation of software metrics data by anyone in your organisation ?

☐ yes ☐ no ☐ don't know

36. In your organisation who do you think is most likely to manipulate metrics data ?

☐ development staff ☐ managers ☐ managers & development staff ☐ no one ☐ don't know

37. Who, in your organisation, decides on what metrics are going to be collected ?

☐ development staff ☐ managers ☐ managers & development staff ☐ don't know

38. Did the initial introduction of software metrics meet with any resistance from :

☐ development staff ☐ managers ☐ managers & development staff ☐ no one ☐ don't know

39. Overall, how successful do you think that your organisation's metrics collection scheme has been ?

☐ very successful ☐ moderately successful ☐ neither successful nor unsuccessful
☐ moderately unsuccessful ☐ very unsuccessful ☐ no metrics collected

Section Seven : Additional Comments

Use the rest of this page and the next page to add any comments that you have. Comments on the following areas would be particularly helpful :

Particular advantages metrics usage has given you and/or your organisation ;
Problems encountered introducing and using metrics;
Lessons learnt about metrics;
Plans you/your organisation has for metrics in the future;
Any comments that you may have about this questionnaire.

Comments

Comments

Thank you for your invaluable help in completing this rather arduous questionnaire.

Please give your name and address below if you are prepared to be contacted in connection with this study :

Name : ____________________ Position : ____________________

Company Address : ____________________

____

____

____

____

Telephone No : ____________________

Email : ____________________

Fax : ____________________

STAFFORDSHIRE UNIVERSITY

SCHOOL OF COMPUTING

METRICS RESEARCH PROJECT

ORGANISATIONAL DETAIL

Name of Organisation : ....

Name of Representative : ....

Job Title : ....

Date : ....

Conducted By ....

Tracy Hall

Size Questions

1. How many staff are employed by the whole organisation ?

☐ a. < 10 ☐ b. 10 - 50 ☐ c. 50 - 100 ☐ d. 100 - 200 ☐ e. 200 +

2. How many development staff are employed by the whole organisation ?

☐ a. < 10 ☐ b. 10 - 50 ☐ c. 50 - 100 ☐ d. 100 - 200 ☐ e. 200 +

3. How old is the organisation ?

☐ a. < 5 years ☐ b. 5 - 10 years ☐ c. 10 - 20 years ☐ d. 20 + years

4. How much effort (as a percentge of the overall organisational effort) goes into ...

	0-10 a	10-20 b	20-30 c	30-50 d	50-70 e	70-100 f
i. In-house development	☐	☐	☐	☐	☐	☐
ii. External development	☐	☐	☐	☐	☐	☐

5. What size project groups do development staff usually work in ?

☐ a. 1 person ☐ b. 2 - 5 ☐ c. 6 - 10 ☐ d. 10 - 20 ☐ e. 20 +

6. How long is a typical software project's duration ?

☐ a. < 1 month ☐ b. 1 - 6 months ☐ c. 6 - 12 months ☐ d. > 12 months

7. How much effort (as a percentge of the overall) goes into ...

	0-10 a	10-20 b	20-30 c	30-50 d	50-70 e	70-100 f
i. New Development	☐	☐	☐	☐	☐	☐
ii. Maintenance	☐	☐	☐	☐	☐	☐
iii. User Support	☐	☐	☐	☐	☐	☐
iv. Other	☐	☐	☐	☐	☐	☐

Applications Questions

8. What kind of applications are developed (as a percentage of overall development) ?

	0-25 a	25-50 b	50-75 c	75-100 d
i. Real Time	☐	☐	☐	☐
ii. Safety Critical	☐	☐	☐	☐
iii. Commercial/Business	☐	☐	☐	☐
iv Other	☐	☐	☐	☐

Development Questions

9. Which languages are used (as a percentage of overall development) ?

	0-25 a	25-50 b	50-75 c	75-100 d
i. COBOL	☐	☐	☐	☐
ii. ADA/Pascal/Modula-4	☐	☐	☐	☐
iii. C/C++	☐	☐	☐	☐
iv. Assembler	☐	☐	☐	☐
v. SQL	☐	☐	☐	☐
vi. Other	☐	☐	☐	☐

10. Which methodologies are used ?

	Yes a	No b
i. SSADM	☐	☐
ii. JSD/JSP	☐	☐
iii. Prototyping	☐	☐
iv. Other	☐	☐

11. Which tools are used ?

	Yes a	No b
i CASE	☐	☐
ii. Z/VDM	☐	☐
iii. Function Points	☐	☐
iv. Project Management	☐	☐
v. Other	☐	☐

Project Management

12. Is the software that is produced certified ?

	Yes a	No b
i. ISO9001	☐	☐
ii. BS5750	☐	☐
iii. TICKIT	☐	☐
iv. Other	☐	☐

13. Are any of the following formal systems used ?

	Always a	Almost Always b	Occasionally c	Never d
i. Documentation Standards	☐	☐	☐	☐
ii. Programming Standards	☐	☐	☐	☐
iii. Test Standards	☐	☐	☐	☐
iv. Configuration Management	☐	☐	☐	☐
v. Specification Reviews	☐	☐	☐	☐
vi. Design Reviews	☐	☐	☐	☐
vii. Code Reviews	☐	☐	☐	☐
viii. Documentation Reviews	☐	☐	☐	☐

14. Does the organisation have a ...

	yes a	no b	Don't Know c
i. Quality Manual	☐	☐	☐
ii. Quality Manager	☐	☐	☐
iii. Metrics Collection Scheme	☐	☐	☐
iv. Metrics Database	☐	☐	☐

15. Does the organisation keep any of the following ?

	Yes a	No b	Don't Know c	Usefulness Rating (1-5
General				
i. Size estimates	☐	☐	☐	☐
ii. Actual size data	☐	☐	☐	☐
iii. Cost estimates	☐	☐	☐	☐
iv. Actual cost data	☐	☐	☐	☐
v. Effort estimates	☐	☐	☐	☐
vi. Actual effort data	☐	☐	☐	☐
vii. Inspection data	☐	☐	☐	☐
Code Related				
viii. Module Interface data	☐	☐	☐	☐
ix. Complexity measures	☐	☐	☐	☐
x. Lines of Code data	☐	☐	☐	☐
xi. Other Module data	☐	☐	☐	☐
Fault Related				
xii. Fault rates	☐	☐	☐	☐
xiii. Defect densities	☐	☐	☐	☐
xiv. Change data	☐	☐	☐	☐
xv. Other	☐	☐	☐	☐

16. What does the lowest level of statistical detail relate to ?

☐ a. Projects ☐ b. Teams ☐ c. Developers ☐ d. Other

17. Who has access to statistical data ?

	yes a	no b	Don't Know c
i. Senior Mgt	☐	☐	☐
ii. Middle Mgt	☐	☐	☐
iii. All Staff	☐	☐	☐
iv. Other	☐	☐	☐

18. What is the organisation's statistical (metric based) data used for ?

	Always a	Usually b	Sometimes c	Never d
i. Assessing Quality	☐	☐	☐	☐
ii. Estimating time/effort	☐	☐	☐	☐
iii. Project Management	☐	☐	☐	☐
iv. Tracking costs	☐	☐	☐	☐
v. Assessing tools/methodologies	☐	☐	☐	☐
vi. Assessing team performance	☐	☐	☐	☐
vii. Assessing staff performance	☐	☐	☐	☐
viii. Other	☐	☐	☐	☐

19. How much effort (as a percentage of the whole effort time) is spent on ...

	< 5% a	6-10 b	10-20 c	20-50 d	50 + e
i. Collecting metric's data	☐	☐	☐	☐	☐
ii. Quality related activity (excluding metrics)	☐	☐	☐	☐	☐

ORGANISATIONAL ISSUES

20. Does the organisation have a PRP scheme ?

☐ a. Yes ☐ b. No

21. How many layers of management is there, in the organisation, between a clerk and the Managing Director ?

☐ a. < 4 ☐ b. 5-7 ☐ c. 8-10 ☐ d. 10-15 ☐ e. 15 +

THE IMPLEMENTATION OF SOFTWARE METRICS

A SURVEY OF CURRENT INDUSTRIAL PRACTICE

SOFTWARE METRICS

A SURVEY OF CURRENT INDUSTRIAL PRACTICE

This questionnaire is designed to allow you to describe how software metrics are being used in your organisation, how software metrics were introduced into your working environment and how effective, you feel, the use of software metrics is.

Please return completed questionnaires to :

Tracy Hall
School of Computing
Staffordshire University
Beaconside
Stafford ST18 OAD

Instructions

In order for the results of this survey to be useful it is most important that you answer this questionnaire as frankly as possible.

1. Please answer all questions in order

2. Most questions require the selection of one answer from several possible answers. Tick the relevant box to make your selection. Please select only one answer.

Some useful definitions :

SOFTWARE METRICS : The measurement of characteristics exhibited by software. For example, the counting of lines of code can be one way that a piece of software's size can be characterised.

ORGANISATION : The firm or company that you work for.

PLEASE BE ASSURED THAT ALL ANSWERS WILL BE TREATED IN THE STRICTEST OF CONFIDENCE

About Yourself

1. How long have you worked for your current organisation ?

☐ up to one year
☐ one year up to five years
☐ five years up to ten years
☐ ten years or more

2. How old are you ?

☐ Less than 25
☐ 25 - 34
☐ 35 - 44
☐ 45 - 54
☐ Over 54

3. Are you ?

☐ Female
☐ Male

4. What is the highest qualification that you have ?

☐ CSE/GCE/GCSE's
☐ 'A' Levels
☐ HNC/HND
☐ BA/BSc
☐ MSc/MA
☐ PhD
☐ Other (Please specify ....)

About Your Job

5. Which one of the following best describes your current job ?

☐ Analyst
☐ Programmer
☐ Analyst/Programmer
☐ Senior/chief Analyst
☐ Senior/chief Programmer
☐ Tester
☐ Manager
☐ Quality Assurer/Controller
☐ Other (Please specify ....)

6. Do you supervise others in your job ?

☐ yes
☐ no

7 How frequently do you engage in each of the following activities ?

	Never	Occasionally	Frequently
Project Management	☐	☐	☐
Systems Analysis	☐	☐	☐
System Design	☐	☐	☐
Programming	☐	☐	☐
Testing	☐	☐	☐
Maintenance	☐	☐	☐
Quality Assurance	☐	☐	☐

8 How frequently do <u>you</u> use the following standards ?

	Never	Occasionally	Frequently
Documentation standards	☐	☐	☐
Programming standards	☐	☐	☐
Test standards	☐	☐	☐
Configuration management	☐	☐	☐
Other	☐	☐	☐

If "other", please specify ....

9 How frequently are <u>you</u> involved in any of the following :

	Never	Occasionally	Frequently
Specification Inspections/Reviews	☐	☐	☐
Design Inspections/Reviews	☐	☐	☐
Code Inspections/Reviews	☐	☐	☐
Document Inspections/Reviews	☐	☐	☐
Quality Manual	☐	☐	☐

About Your Organisation

10. Does your organisation allow you to ...

	Always	Usually	Rarely	Never
Work efficiently	☐	☐	☐	☐
Work effectively	☐	☐	☐	☐
Work flexibly	☐	☐	☐	☐
Work autonomously	☐	☐	☐	☐
Produce high quality work	☐	☐	☐	☐

11 Different organisations have different goals. Are the goals of your organisation made clear to you ?

☐ Always ☐ Usually ☐ Rarely ☐ Never

12 You will find below a list of 5 possible software development goals. Please rank these goals in two different ways. Firstly, rank the goals according to what you understand your organisation's goals to be. Secondly, rank the goals according to your own preferred goals.

(Please put numbers 1 to 5 in the boxes provided - 1 representing the most important goal and 5 representing the least important goal)

	Organisation's Goals	Your Goals
Speed of delivery	☐	☐
Reliability of software	☐	☐
Closeness of software to the specification	☐	☐
Minimal costs incurred	☐	☐
Satisfaction of user	☐	☐

13 How satisfied are you with your ...

	Very satisfied	Satisfied	Dissatisfied	Very dissatisfied
salary ?	☐	☐	☐	☐
working conditions ?	☐	☐	☐	☐
working environment ?	☐	☐	☐	☐
manager ?	☐	☐	☐	☐
training ?	☐	☐	☐	☐
job generally ?	☐	☐	☐	☐

14 Do you receive enough feedback and information on...

	Yes	No
How your organisation is performing ?	☐	☐
How your section/team is performing ?	☐	☐
How you are performing ?	☐	☐

15 How much criticism, by employees, is there about your organisation ?

None	Very little	Some	Quite a lot	A great deal
☐	☐	☐	☐	☐

About Software Metrics

16 Does your organisation have ...

	Yes	No	Don't Know
a metrics collection scheme ?	☐	☐	☐
a metrics database ?	☐	☐	☐
a quality manual ?	☐	☐	☐
procedures to ensure quality ?	☐	☐	☐
a quality manager ?	☐	☐	☐
BS5750 certification ?	☐	☐	☐
Documentation standards ?	☐	☐	☐
Programming standards ?	☐	☐	☐
Test standards ?	☐	☐	☐
a configuration management system ?	☐	☐	☐
Specification Reviews ?	☐	☐	☐
Design Inspections/Reviews ?	☐	☐	☐
Code Inspections/Reviews ?	☐	☐	☐
Documentation Inspections/Reviews	☐	☐	☐

17 Does your organisation record any of the following information ?

Please also tick the last box if you are directly involved in the recording of this information.

Analysis Data	Yes	No	Don't Know	Involved personally
Function points	☐	☐	☐	☐
Size estimates	☐	☐	☐	☐
Cost estimates	☐	☐	☐	☐
Effort estimates	☐	☐	☐	☐
Inspection data	☐	☐	☐	☐

Design Data:	Yes	No	Don't Know	Involved personally
Design review data	☐	☐	☐	☐
Effort data	☐	☐	☐	☐

Code Data:	Yes	No	Don't Know	Involved personally
Module Interface data	☐	☐	☐	☐
Complexity measures	☐	☐	☐	☐
Lines of Code data	☐	☐	☐	☐
Other Module data	☐	☐	☐	☐
Effort data	☐	☐	☐	☐
Inspection data	☐	☐	☐	☐

Test Data:	Yes	No	Don't Know	Involved personally
Fault rates (Time between faults)	☐	☐	☐	☐
Defect densities (Faults per LoC)	☐	☐	☐	☐
Change data	☐	☐	☐	☐
Effort data	☐	☐	☐	☐
Review data	☐	☐	☐	☐

If any other similar information is collected in your organisation then please attach details.

What is Your Opinion of software Metrics

18 How accurately is statistical information about software recorded in your organisation ?

very accurately	quite accurately	quite inaccurately	very inaccurately	don't know
☐	☐	☐	☐	☐

19 How much time, on average, do you spend recording statistical data during your working week ?

less than 1 hour	1 - 5 hours	6 - 10 hours	Over 10 hours
☐	☐	☐	☐

20 How useful an activity do you consider the collection of statistical data, about software, to be ?

very useful	quite useful	not very useful	not useful at all	don't know
☐	☐	☐	☐	☐

21. Does your organisation provided any feedback on the data that has been collected to the people who recorded the data ?

☐ yes ☐ no ☐ don't know

If "yes" please give brief details ....

22. Does your organisation use the data that has been collected ...

	yes	no	don't know
constructively	☐	☐	☐
responsibly	☐	☐	☐
secretly	☐	☐	☐
to assess peoples' performance	☐	☐	☐
to assess the organisation's performance	☐	☐	☐
to determine people's pay	☐	☐	☐
to assess the quality of software	☐	☐	☐
to assess the software development process	☐	☐	☐
as a costing tool	☐	☐	☐

23. Who, in your organisation, decides what statistical data is going to be collected about software ?

managers	development staff	managers & development staff	other	don't know
☐	☐	☐	☐	☐

If "other" please specify ....

7

24. Do you consider that your organisation collects the right data about software ?

yes ☐ no ☐ don't know ☐

25. Do you think that there is any manipulation of software statistics by anyone in your organisation ? For example people claiming that more time was spent on doing something than was actually the case.

yes ☐ no ☐ don't know ☐

What You Think of this Questionnaire !

26. Were the instructions clear enough ?

☐ Yes ☐ No

27. How many of the questions were ...

	All	Most	None
clear	☐	☐	☐
relevant	☐	☐	☐
intrusive	☐	☐	☐
interesting to answer	☐	☐	☐
patronising	☐	☐	☐

28. How well did you feel that this questionnaire allowed you to describe your feelings about software metrics in your organisation ?

☐ Very well ☐ Adequately ☐ Not at all

Any specific comments that you may have about the questionnaire will be gratefully received.

PLEASE RETURN THIS QUESTIONNAIRE, AS SOON AS POSSIBLE, TO :

TRACY HALL,
SCHOOL OF COMPUTING,
STAFFORDSHIRE UNIVERSITY,
BEACONSIDE,
STAFFORD, ST18OAD.

Appendix Four

Sample SPSS output

4.1 An example SPSS programme

The following programme produces frequency tables for all the data contained in file *Questionnaire.dat.*

```
SET LISTING = 'A:       .LIS'

 DATA LIST FILE = 'A:Questionnaire.DAT'
     / Q1 4 (A)
    Q2 5 (A)
    Q3 6 (A)
    Q4 7 (A)
    Q5 8 (A)
    Q6 9 (A)
    Q7i 10 (A)  Q7ii 11 (A)  Q7iii 12 (A)  Q7iv 13 (A)  Q7v 14 (A)
            Q7vi 15 (A)  Q7vii 16 (A)
    Q8i 17 (A) Q8ii 18 (A) Q8iii 19 (A) Q8iv 20 (A) Q8v 21 (A)
    Q9i 22 (A) Q9ii 23 (A) Q9iii 24 (A) Q9iv 25 (A)
    Q10i 26 (A) Q10ii 27 (A) Q10iii 28 (A) Q10iv 29 (A) Q10v 30 (A)
    Q11 31 (A)
    Q12ia 32 Q12ib 33 Q12iia 34 Q12iib 35 Q12iiia 36 Q12iiib 37
            Q12iva 38 Q12ivb 39 Q12va 40 Q12vb 41
    Q13i 42 (A) Q13ii 43 (A) Q13iii 44 (A) Q13iv 45 (A) Q13v 46 (A)
            Q13vi 47 (A)
    Q14i 48 (A) Q14ii 49 (A) Q14iii 50 (A)
    Q15 51 (A)
    Q16i 52 (A) Q16ii 53 (A) Q16iii 54 (A) Q16iv 55 (A) Q16v 56 (A)
            Q16vi 57 (A) Q16vii 58 (A) Q16viii 59 (A) Q16ix 60 (A)
            Q16x 61 (A) Q16xi 62 (A) Q16xii 63 (A) Q16xiii 64 (A)
            Q16xiv 65 (A)
    Q171i 66 (A) Q171ii 68 (A) Q171iii 70 (A) Q171iv 72 (A) Q171v 74 (A)
    Q171id 67 (A) Q171iid 69 (A) Q171iiid 71 (A) Q171ivd 73 (A) Q171vd 75 (A)
    Q172i 76 (A) Q172ii 78 (A)
    Q172id 77 (A) Q172iid 79 (A)
    Q173i 80 (A) Q173ii 82 (A) Q173iii 84 (A) Q173iv 86 (A)
            Q173v 88 (A) Q173vi 90 (A)
    Q173id 81 (A) Q173iid 83 (A) Q173iiid 85 (A) Q173ivd 87 (A)
            Q173vd 89 (A) Q173vid 91 (A)
    Q174i 92 (A) Q174ii 94 (A) Q174iii 96 (A) Q174iv 98 (A) Q174v 100 (A)
    Q174id 93 (A) Q174iid 95 (A) Q174iiid 97 (A) Q174ivd 99 (A) Q174vd 101 (A)
    Q18 102 (A)
    Q19 103 (A)
    Q20 104 (A)
    Q21 105 (A)
    Q22i 106 (A) Q22ii 107 (A) Q22iii 108 (A) Q22iv 109 (A) Q22v 110 (A)
            Q22vi 111 (A) Q22vii 112 (A) Q22viii 113 (A) Q22ix 114 (A)
    Q23 115 (A)
    Q24 116 (A)
    Q25 117 (A)
    Q26 118 (A)
    Q27 119 (A)
    Q28i 120 (A) Q28ii 121 (A) Q28iii 122 (A) Q28iv 123 (A) Q28v 124 (A)
    Q29 125 (A).
```

```
VARIABLE LABELS
     Q1 'Service Length'
     Q2 'Age'
     Q3 'Sex'
     Q4 'Qualifications'
     Q5 'Job Description'
     Q6 'Staff Supervisor ?'
     Q7i 'Project Manage ?'  Q7ii 'Systems Analysis ?'  Q7iii 'Systems Design ?'
        Q7iv 'Programming ?'  Q7v 'Testing ?' Q7vi 'Maintenance ?'
        Q7vii 'QA ?'
     Q8i 'Documentation Standards ?' Q8ii 'Programming Standards ?'
        Q8iii 'Test Standards ?' Q8iv 'Configuration Mgt Stds ?'
        Q8v 'Other'
     Q9i 'Spec Reviews ?' Q9ii 'Design Reviews ?' Q9iii 'Code Reviews ?'
        Q9iv 'Documentation Reviews ?'
     Q10i 'Work Efficiently ?' Q10ii 'Work Effectively ?' Q10iii 'Work Flexibly ?'
        Q10iv 'Work Autonomously' Q10v 'Produce High Quality Work ?'
     Q11 'How Clear are Org Goals ?'
     Q12ia 'Org goal = Speed' Q12ib 'Your goal = Speed'
       Q12iia 'Org goal = Reliability' Q12iib 'Your goal = Reliability'
       Q12iiia 'Org goal = Conformance'
       Q12iiib 'Your goal = Conformance' Q12iva 'Org goal = Low Costs'
       Q12ivb 'Your goal = Low Costs' Q12va 'Org goal = User Satisfaction'
       Q12vb 'Your goal = User Satisfaction'
     Q13i 'Salary Satisfaction' Q13ii 'Working Conditions Satisfaction'
       Q13iii 'Working Environment Satisfaction' Q13iv 'Manager Satisfaction'
       Q13v 'Training Satisfaction' Q13vi 'General Job Satisfaction'
     Q14i 'Organisational Feedback ?' Q14ii 'Team Feedback ?'
       Q14iii 'Individual Feedback ?'
     Q15 'Amount of critism about organisation ?'
     Q16i 'A Metrics Scheme ?' Q16ii 'A Metrics DB ?' Q16iii 'A QA Manual ?'
       Q16iv 'Procedures for quality ?' Q16v 'A Qaulity Manager ?'
       Q16vi 'BS5750 ?' Q16vii 'Documentation Standards ?'
       Q16viii 'Programming Standards ?' Q16ix 'Test Standards ?'
       Q16x 'Configuration Management ?' Q16xi 'Spec Reviews ?'
       Q16xii 'Design Reviews ?' Q16xiii 'Code Reviews ?' Q16xiv 'Doc Reviews'
     Q171i 'Org Collects Function Points' Q171ii 'Org Collects Size Estimates'
       Q171iii 'Org Collects Cost Estimates' Q171iv 'Org Collects Effort Ests'
       Q171v 'Org Collects Inspection Data'
     Q171id 'Personally Collect Function Points'
       Q171iid 'Personally Collect Size Ests'
       Q171iiid 'Personally Collect Cost Ests'
       Q171ivd 'Personally Collect Effort Ests'
       Q171vd 'Personally Collect Inspection Data (Analysis)'
     Q172i 'Org Collects Design Review Data'
       Q172ii 'Org Collects Design Effort Data'
     Q172id 'Personally Collect Design Review Data'
       Q172iid 'Personally Collect Design Effort Data'
     Q173i 'Org Collects Interface Data' Q173ii 'Org Collects Complexity Data'
       Q173iii 'Org Collects LoC Data' Q173iv 'Org Collects Other Module Data'
       Q173v 'Org Collects Coding Effort Data'
       Q173vi 'Org Collects Code Inspection Data'
     Q173id 'Personally Collect Interface Data'
       Q173iid 'Personally Collect Complexity Data'
       Q173iiid 'Personally Collect LoC Data'
       Q173ivd 'Personally Collect Other Module Data'
       Q173vd 'Personally Collect Coding Effort Data'
       Q173vid 'Personally Collect Code Inspection Data'
     Q174i 'Org Collects Fault Rates' Q174ii 'Org Collects Defect Densities'
       Q174iii 'Org Collects Change Data'
       Q174iv 'Org Collects Testing Effort Data'
       Q174v 'Org Collects Test Review Data'
     Q174id 'Personally Collect Fault Rates'
       Q174iid 'Personally Collect Defect Densities'
       Q174iiid 'Personally Collect Change Data'
       Q174ivd 'Personally Collect Testing Effort Data'
       Q174vd 'Personally Collect Test Review Data'
     Q18 'Accuracy of Measurements ?'
     Q19 'Time Spent Collection Data'
     Q20 'Metrics Usefulness ?'
     Q21 'Feedback on metrics given ?'
     Q22i 'Data Used Constructively ?' Q22ii 'Data Used Responsibly ?'
      Q22iii 'Data Used Secretly ?' Q22iv 'Data Used for Individual Assessment ?'
      Q22v 'Data Used for Organisational Assessment ?'
      Q22vi 'Data Used as Salary Basis ?' Q22vii 'Data Used for SW Quality ?'
      Q22viii 'Data Used for SW Process ?' Q22ix 'Data Used for Costing ?'
     Q23 'Who designs data to collect ?'
     Q24 'Is Right Data Collected ?'
     Q25 'Any Manipulation of Figures ?'
     Q26 'Who Manipulates Figures ?'.
```

```
VALUE LABELS
   Q1 'a' 'Up to 1 Year' 'b' '1 to 5 Years' 'c' '5 to 10 Years'
          'd' 'Over 10 Years'
   /Q2 'a' '< 25' 'b' '25 to 34' 'c' '35 to 44' 'd' '45 to 54' 'e' 'Over 54'
   /Q3 'a' 'Female' 'b' 'Male'
   /Q4 'a' 'GCSEs' 'b' 'A Levels' 'c' 'HND' 'd' 'Degree' 'e' 'MSc'
          'f' 'PhD' 'g' 'Other'
   /Q5 'a' 'Analyst' 'b' 'Programmer' 'c' 'Analyst/Programmer'
          'd' 'Senior Analyst' 'e' 'Senior Programmer' 'f' 'Tester'
          'g' 'Manager' 'h' 'QA' 'i' 'Other'
   /Q7i Q7ii Q7iii Q7iv Q7v Q7vi Q7vii Q8i Q8ii Q8iii Q8iv Q8v Q9i Q9ii Q9iii
          Q9iv 'a' 'Never' 'b' 'Occasionally' 'c' 'Frequently'
   /Q10i Q10ii Q10iii Q10iv Q10v Q11 'a' 'Always' 'b' 'Usually' 'c' 'Rarely'
          'd' 'Never'
   /Q13i Q13ii Q13iii Q13iv Q13v Q13vi 'a' 'Very Satisfied' 'b' 'Satisfied'
          'c' 'Dissatisfied' 'd' 'Very Dissatisfied'
   /Q15 'a' 'None' 'b' 'Very Little' 'c' 'Some' 'd' 'Quite a Lot'
          'e' 'A Great Deal'
   /Q6 Q14i Q14ii Q14iii Q16i Q16ii Q16iii Q16iv Q16v Q16vi Q16vii Q16viii
          Q16ix Q16x Q16xi Q16xii Q16xiii Q16iv Q171i Q171ii Q171iii Q171iv
          Q171v Q171id Q171iid Q171iiid Q171ivd Q171vd Q172i Q172ii Q172id
          Q172iid Q173i Q173ii Q173iii Q173iv Q173v Q173vi Q173id Q173iid
          Q173iiid Q173ivd Q173vd Q173vid Q174i Q174ii Q174iii Q174iv Q174v
          Q174id Q174iid Q174iiid Q174ivd Q174vd Q21 Q22i Q22ii Q22iii
          Q22iv Q22v Q22vi Q22vii Q22viii Q22ix Q24 'a' 'Yes' 'b' 'No'
          'c' 'Dont Know'
   /Q18 'a' 'Very Accuarately' 'b' 'Quite Accurately' 'c' 'About Right'
          'd' 'Quite Inaccurately' 'e' 'Very Inaccurately' 'f' 'Dont Know'
   /Q19 'a' '< 1 Hour' 'b' '1 to 5 Hours' 'c' '6 to 10 Hours' 'd' 'Over 10 Hours'
   /Q20 'a' 'Very Useful' 'b' 'Quite Useful' 'c' 'Not Very Useful'
          'd' 'Not Useful at all' 'e' 'Dont Know'
   /Q23 'a' 'Managers' 'b' 'Developers' 'c' 'Managers and Developers'
          'd' 'Other' 'e' 'Dont Know'
   /Q25 'a' 'Often' 'b' 'Occasionally' 'c' 'Never' 'd' 'Never' 'e' 'Dont Know'
   /Q26 'a' 'Developers' 'b' 'Managers' 'c' 'Neither' 'd' 'Both' 'e' 'Dont Know'


MISSING VALUE
   Q1 ('z') Q2 ('z') Q3 ('z') Q4 ('z') Q5 ('z') Q6 ('z') Q7i ('z')
   Q7ii ('z') Q7iii ('z') Q7iv ('z') Q7v ('z') Q7vi ('z') Q7vii ('z')
   Q8i ('z') Q8ii('z') Q8iii ('z') Q8iv ('z') Q8v ('z') Q9i ('z')
   Q9ii ('z') Q9iii ('z') Q9iv ('z') Q10i ('z') Q10ii ('z') Q10iii ('z')
   Q10iv ('z') Q10v ('z') Q11 ('z') Q13i ('z') Q13ii ('z') Q13iii ('z')
   Q13iv ('z') Q13v ('z') Q13vi ('z') Q14i ('z') Q14ii ('z') Q14iii ('z')
   Q15 ('z') Q16i ('z') Q16ii ('z') Q16iii ('z') Q16iv ('z') Q16v ('z')
   Q16vi ('z') Q16vii ('z') Q16viii ('z') Q16ix ('z') Q16x ('z') Q16xi ('z')
   Q16xii ('z') Q16xiii ('z') Q16xiv ('z') Q171i ('z') Q171ii ('z')
   Q171iii ('z') Q171iv ('z') Q171v ('z') Q171id ('z') Q171iid ('z')
   Q171iiid ('z') Q171ivd ('z') Q171vd ('z') Q172i ('z') Q172ii ('z')
   Q172id ('z') Q172iid ('z') Q173i ('z') Q173ii ('z') Q173iii ('z')
   Q173iv ('z') Q173v ('z') Q173vi ('z') Q173id ('z') Q173iid ('z')
   Q173iiid ('z') Q173ivd ('z') Q173vd ('z') Q173vid ('z') Q174i ('z')
   Q174ii ('z') Q174iii ('z') Q174iv ('z') Q174v ('z') Q174id ('z')
   Q174iid ('z') Q174iiid ('z') Q174ivd ('z') Q174vd ('z') Q18 ('z')
   Q19 ('z') Q20 ('z') Q21 ('z') Q22i ('z') Q22ii ('z') Q22iii ('z')
   Q22iv ('z') Q22v ('z') Q22vi ('z') Q22vii ('z') Q22viii ('z')
   Q22ix ('z') Q23 ('z') Q24 ('z') Q25 ('z') Q26 ('z') Q27 ('z')
   Q28i ('z') Q28ii ('z') Q28iii ('z') Q28iv ('z') Q28v ('z')
   Q29 ('Z').


FREQUENCIES VARIABLES = ALL.
```

4.2 Example frequency tables

Following are examples of some of the frequency tables that the SPSS programme in 4.1 will produce.

Q1 Service Length

Value Label	Value	Frequency	Percent	Valid Percent	Cum Percent
Up to 1 Year	a	5	2.8	2.8	2.8
1 to 5 Years	b	31	17.4	17.6	20.5
5 to 10 Years	c	63	35.4	35.8	56.3
Over 10 Years	d	77	43.3	43.8	100.0
	z	2	1.1	Missing	
	Total	178	100.0	100.0	

Valid cases 176 Missing cases 2

- -

Q2 Age

Value Label	Value	Frequency	Percent	Valid Percent	Cum Percent
< 25	a	15	8.4	8.4	8.4
25 to 34	b	73	41.0	41.0	49.4
35 to 44	c	68	38.2	38.2	87.6
45 to 54	d	21	11.8	11.8	99.4
Over 54	e	1	.6	.6	100.0
	Total	178	100.0	100.0	

Valid cases 178 Missing cases 0

- -

Q4 Qualifications

Value Label	Value	Frequency	Percent	Valid Percent	Cum Percent
GCSEs	a	23	12.9	12.9	12.9
A Levels	b	50	28.1	28.1	41.0
HND	c	24	13.5	13.5	54.5
Degree	d	66	37.1	37.1	91.6
MSc	e	4	2.2	2.2	93.8
PhD	f	2	1.1	1.1	94.9
Other	g	9	5.1	5.1	100.0
	Total	178	100.0	100.0	

Valid cases 178 Missing cases 0

- -

Q5 Job Description

Value Label	Value	Frequency	Percent	Valid Percent	Cum Percent
Analyst	a	1	.6	.6	.6
Programmer	b	31	17.4	17.4	18.0
Analyst/Programmer	c	40	22.5	22.5	40.4
Senior Analyst	d	12	6.7	6.7	47.2
Senior Programmer	e	18	10.1	10.1	57.3
Tester	f	5	2.8	2.8	60.1
Manager	g	40	22.5	22.5	82.6
QA	h	9	5.1	5.1	87.6
Other	i	22	12.4	12.4	100.0
	Total	178	100.0	100.0	

Valid cases 178 Missing cases 0

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Q6        Staff Supervisor ?

                                                            Valid     Cum
Value Label                    Value  Frequency  Percent  Percent  Percent

Yes                                a         96     53.9     54.2     54.2
No                                 b         81     45.5     45.8    100.0
                                   z          1       .6  Missing
                                        -------  -------  -------
                               Total        178    100.0    100.0

Valid cases      177      Missing cases        1
```

4.3 An example crosstabulation

The following SPSS command:

```
CROSSTABS        /TABLES Q21 by Q20
                 /CELLS
                 /STATISTICS = CHISQ
                 /FORMAT NOBOX.
```

Produces the crosstabulation:

```
Q21  Feedback on metrics given ?  by  Q20  Metrics Usefulness ?

                     Q20                                        Page 1 of 1
             Count  |
           Row Pct  |Very Use Quite Us Not Very Not Usef Dont Kno
           Col Pct  |ful       eful      Useful  ul at al w           Row
           Tot Pct  |    a        b        c        d        e       Total
Q21        ---------+--------------------------------------------
                 a  |    13       22        2        2                 39
  Yes               |  33.3     56.4      5.1      5.1               22.2
                    |  34.2     25.3      5.9     22.2
                    |   7.4     12.5      1.1      1.1
                    |
                 b  |    13       29       17        5        4        68
  No                |  19.1     42.6     25.0      7.4      5.9      38.6
                    |  34.2     33.3     50.0     55.6     50.0
                    |   7.4     16.5      9.7      2.8      2.3
                    |
                 c  |    12       36       15        2        4        69
  Dont Know         |  17.4     52.2     21.7      2.9      5.8      39.2
                    |  31.6     41.4     44.1     22.2     50.0
                    |   6.8     20.5      8.5      1.1      2.3
                    |
            Column       38       87       34        9        8       176
             Total     21.6     49.4     19.3      5.1      4.5     100.0

        Chi-Square                  Value            DF           Significance
--------------------           -----------          ----          ------------

Pearson                           13.38686            8                .09921
Likelihood Ratio                  16.45576            8                .03630
```

Appendix Five

Published Papers

1. Journal papers

Wilson D, Hall T (1998) "Software Quality perceptions: A Pilot Study"
Software Quality Journal, Mar, vol 7(1), pp67-75

Hall T, Wilson D (1997) "Views of Software Quality: A Field Report"
IEE Procs on Software Engineering, April, pp111-118

Hall T, Fenton N (1997) "Implementing Effective Software Metrics Programmes"
IEEE Software, March, pp55-65

Hall T, Fenton N (1996) "Software Quality Programmes: A Snapshot of Theory versus Reality"
SoftwareQuality Journal, Dec, pp235-242

Hall T, Fenton N (1994) "Implementing Software Metrics: the Critical Success Factors"
Software Quality Journal, Dec, pp195-208

2. Invited papers

Hall T (1997) "The Quality of Metrics: Experiences of Practitioners"
Invited Paper, *Centre for Software Reliability Workshop*, Reading, Mar

Hall T, Fenton N (1996) "Implementing Successful Software Metrics Programmes"
Invited Paper, *Bournemouth Metrics Symposium*, Bournemouth University, April

3. Chapters in books

Hall T (1996) "Evaluating Software Quality Mechanisms - A Snapshot Picture"
in *Improving Quality,* Bray M, Ross M, Staples G (Eds), Computational Mechanics Publication (Procs of *BCS Software Quality Management Conf,* Cambridge, April), pp251-261

Hall T, Wilson D (1997) "The Real State of Software Quality: Practitioners' Experiences"
in *The Quality Challenge*, Hawkins C, Ross M, Staples G (Eds), Mechanical Engineering Publications (Procs of *BCS Software Quality Management* Conf., Bath, Mar), pp179-195

Hall T (1995) "What do Developers really think about Software Quality?"
in *Quality Management,* Ross M *et al* (Eds), Computational Mechanics Publication (Procs of BCS *Software Quality Management Conference*, Seville, April), pp359-369

4. Conference papers

Hall T, Fenton N (1995) "Software Practitioners and Software Quality Improvement"
ASQC 6th International Conference on Software Quality, Austin, Texas, Nov, pp313-325

Hall T, Fenton N (1994) "What do Developers really Think about Software Metrics?"
Applied Software Measurement Conference, California, Nov, pp685-709

Hall T, Fenton N (1994) "Implementing Software Metrics"
Applied Software Measurement Conference, California, Nov, pp247-264

5. Other articles

Hall T (1997) "Implementing Effective Software Measurement Programmes"
CSR Newsletter, May

Hall T (1995) "No Quality without Equality" Quality Time Column
IEEE Software, Mar, pp101-102