Achieving Business Excellence

in

Software Quality Management

By Michael James Elliott

A Doctoral Thesis

submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy of Loughborough University

August 2008

Abstract

Many companies have had difficulties in achieving success with software process improvement initiatives or have had adverse experiences in implementing quality systems. With a plethora of standards available and the numerous frameworks to apply best practice, none appears to act as a panacea to guarantee fulfilment or realise a true Return-on-Investment. This thesis proposes a holistic approach to software process improvement, describing a range of supporting tools and methods highlighting a true understanding of the customer base and associated cultures.

The research aim was to develop and evaluate a demonstrably effective and efficient software quality management methodology suitable for a technical company such as AWE plc. To be effective the methodology must deliver an improved conformance to the quality standards and deliver real process improvement. To be efficient the methodology must deliver a real Return-on-Investment.

Case studies on the implementation of the quality system were carried out at AWE plc. Each case study provided a further opportunity to measure and analyse the success or otherwise of that method or tool for further refinement. Audits, self-assessment, training, system design, marketing, and the people skills associated with a consultation process are all examined in detail.

The research methodology has demonstrated its success as case studies show that steady improvement in implementing the software quality system has occurred year on year. This success has been validated by third party ISO 9001 assessments and has led to an enhancement in AWE plc reputation as a centre of software excellence. The approach has overcome cultural resistance and changed working practices. With a philosophy of customer care, consultation, and active engagement, practitioners now adopt best-practice quality management principles. The cost effectiveness of this methodology means its adoption could be considered by any organisation whether large or small.

Acknowledgements

I would like to thank my supervisors Ray Dawson and Janet Edwards for their advice and support throughout the duration of this research. In particular to Ray for providing me the key to the door that opened a new landscape and the acquisition of the attribute of "outside eyes" even when close to the action.

I would like to thank Graham Wing, a friend and colleague, for his significant contribution at the start of the journey. Perhaps it took longer than planned, however the "outstanding" achievement was met.

To members of the US NNSA/NWC Software Quality Assurance Sub-committee SQAS, I give my thanks for their friendship and support and for providing a platform to present ideas, experiences and convey lessons learned, and for being a focus of motivation and interest somewhat lacking in my work environment over the duration of the research. In particular: Don Schilling, Kansas City Plant, Dave Peercy, Sandia National Labs, Nancy Smith, Savannah River Site, Mike Lacker and Maysa Peterson, Los Alamos National Laboratory, Carolyn Owens, Lawrence Livermore National Laboratory and Kathleen Canal DoE HQ.

Above all, I must thank my Family, my wife Tina and my sons Richard and Andrew for showing a degree of patience throughout the research. Now they can have the computer back!

Table of Contents

Chapter 1 - Introduction

Chapter Preface

This chapter places the research into context by introducing the environment in which the research was conducted. A brief background to AWE plc is conveyed as well as its unique cultures. The coincidental events and motivation that initiated the research and its goals are described, as well as the aim and the detailed objectives to achieve that aim. Finally the thesis outline is provided as a précis of each of the main chapters.

1.1 The Research Environment

The principal environment for this research has been the processes associated with the adoption and use of the software quality management system at AWE plc, and the behaviours of personnel impacted by its rules. Although this could be considered a narrow focus, AWE plc is in fact a very diverse establishment. With the breadth of technologies associated with nuclear weapons production, all the main sciences, physics, chemistry and biology, are researched and developed. All the main engineering subjects, electrical, mechanical, electronic, software and information technology are practised. All phases of the nuclear weapon life cycle are, or at one time have been, undertaken by staff at AWE plc, research and development, design, production, trials and testing of varying sorts, in-service support, disassembly and disposal. With such a broad range and diverse set of work activities this attracts an equally broad and diverse variety of people that set-up and establish their own unique working practices and cultures. The types of people include highly qualified scientists and engineers to frontline tradesmen, from financial accountants to "people types" in human resources.

1.2 AWE plc

In 1950, The Atomic Weapons Research Establishment (AWRE), as it was known then, was originally under the management of the government department, The Ministry of Supply. It was then transferred to the United Kingdom Atomic Energy Authority (UKAEA) in 1954, and subsequently in 1973, the establishment returned to Civil Service Management, under the Ministry of Defence (MoD) Procurement Executive. AWE has been at the forefront of Britain's nuclear deterrent for over fifty years.

In 1993, AWE became a government owned, contractor operated establishment and since then has been managed by two large contractor consortiums. Hunting Engineering, Brown and Root, and The Atomic Energy Authority formed the first consortium trading under the company name of Hunting Brae from 1993 to 2000. From the year 2000 to the present day, Lockheed Martin, Serco and BNFL formed AWE ML (Management Limited) and have managed operations at the establishment. The site management and operations are still monitored at different levels by project teams from the MoD. These are known as Integrated Project Teams (IPTs) and work closely with AWE plc employees. The number of personnel working at AWE has reduced from approximately 6,500 in 1993 to 4,500 in 2004, although recent projects from infrastructure improvements have seen a rise in the number of employees.

1.3 AWE Culture and Organisation

Despite these managerial changes, the prominent organisational arrangements are very much functional line management oriented, with large numbers (hundreds) of staff grouped together to perform the main functional activities as directorates. In many ways this has led to the perception that AWE is run by separate companies within one. Under the initial government owned, contractor operated arrangements the main core functional organisations were: a scientifically oriented research group with highly qualified scientific personnel, a weapons or systems engineering group

with the major concentration of highly qualified engineers, a production technology group which has a mix of qualified engineers and scientific staff, and a site development group providing the site infrastructure and support with a less qualified, but highly skilled engineering workforce. Support activities are also provided by a smaller number of staff in support groups, namely personnel, financial, and commercial. Safety, quality, environment and the security functions were amalgamated in an assurance group.

A key organisational issue that has a significant impact on the software quality management system and the facilitation of this research is the appointment of local area software representatives (LASRs). Each management area that manages software is required by the software quality management system to appoint a LASR to perform a number of software quality related duties. A total of 70 LASRs were appointed across the company, 35 of which were from areas that had significant software. The other 35 were procurers of commercial, off-the-shelf (COTS) software. In these areas, the LASRs task was to keep a control of software use, by registering or recording an inventory of software licences and their respective owners. In a software extensive area, a LASR was also required to approve the local software control plans (SCP) and assign a risk category to each software item or product. The software category defines the control activities required for its subsequent management. There is no consistency to the type of people appointed as LASRs, which varies from administrative, scientific, engineer, developer or staff with other quality system roles. This is due to the fact that the software quality system did not specify any skills or experience for the appointment.

The company has a central Information Technology (IT) group that provides some in-house programming support and contract management of IT service providers covering a range of activities as well as the IT network infrastructure. However, there is no central software engineering support. AWE plc is well known for some world class software in such areas as weapon simulation, weapon effects, engineering design, system control and data acquisition and analysis.

A requirement within the new contract of 1993 was for AWE plc to gain ISO 9001 certification by 1994. This was initially achieved through second party assessments by the Ministry of Defence (MoD). Six monthly assessments took place until 2003 when independent third party certification was achieved.

This research was initially established under the first consortium contract which was focused on maintaining the capability to develop future weapons without actually producing them. This resulted in an element of company downsizing with voluntary redundancies for an initial four-year period. However, a more recent up-turn in funding has seen a ramp-up on personnel numbers and the construction of modern facilities.

1.4 The Research Motivation

The initiating event for the author's research journey was the attendance at the 2001 Software Quality Management (SQM) conference at Loughborough University organised by my supervisor Ray Dawson, and Margaret Ross of Southampton Solent University. By this time some of the problems with the implementation of the SQMS at AWE had been identified. However, these problems had not been expected as AWE would have been considered a relatively mature quality organisation, with ISO 9001 certification having been established and assessed for some five years. Had the author been armed with an understanding of these problems with implementation, a different program of work would have ensued. Part of the author's practice was to investigate various standards and methodology for best practice approaches and for process improvement, and to research various case studies of both successful projects and ones that had failed in some way to gain lessons learned. Part of that quest facilitated a chance Google search on the term "software quality management" that returned the impending Software Quality Management conference at Loughborough. A trawl through the conference agenda conveyed a number of case studies, methodologies and recent research on software quality issues and process

improvement. A call to the course director, Margaret Ross, not only confirmed attendance was still possible but there was also an opportunity to present the author's research on process improvement, as one of the speakers had cancelled.

At the conference the author was indeed exposed to a plethora of process improvement initiatives and research studies conducted by PhD students. It became apparent to the author that he had little time at his workplace to reflect on problems and develop well thought through initiatives that would truly improve working practices and make a more effective and efficient use of time. It was these thought processes and follow-up discussion with Ray Dawson at Loughborough University that presented the opportunity to align the author's own working research with that of a research doctorate. The principal driver would be that this opportunity would formalise the research to form an appropriate methodology to improve working practices in response to the defined software quality management system. This opportunity fitted in well with Loughborough University's own initiatives to provide practical research opportunities under its "links with industry" project led by Ray Dawson.

1.5 Aim and Research Hypothesis

The principal aim of this research is:

To develop and evaluate a demonstrably effective and efficient software quality management methodology suitable for a technical company such as AWE plc. To be effective the methodology must deliver an improved conformance to the quality standards adopted by the company, it must deliver real process improvement to the company, and it must be accepted and willingly used by employees at all levels of the company from the engineer developing the software to the most senior management. To be efficient the methodology must deliver a real Return On Investment (ROI), adding value to the company within a realistic timeframe (ie. months and years rather than decades).

The research hypothesis associated with this aim is that such a methodology is possible to achieve using research techniques based on a literature review and action research at the company.

1.6 Objectives

To achieve the aim of the research the following objectives were set:

1. Conduct a survey of published literature to determine the current state-of-the-art in software quality management with a focus on auditing, self-assessment, training and cost-benefit and to see if there is any best practice methodologies that could be used at AWE plc. This review would also establish the validity of this research and the potential contribution to knowledge that it would make.
2. By inspecting the records of non-conformance raised during audits on software quality management practice, assess the approaches taken in gathering data to identify weaknesses in the process and the impact this has on effectiveness and auditor motivation.
3. Conduct a survey of auditors and auditees and influencing literature, to investigate the reasons and perceptions why audits do not always add value.
4. Develop an improved process model for auditing to overcome the reasons and perceptions why audits do not always add value based on the findings of literature review and survey. This improved process model should be implemented and tested at AWE plc.
5. Determine the extent of practitioner adoption of a software quality management system by devising a measurement system and audit or self-assessment question-set to be utilised on quality assessments at AWE that capture both the level of implementation of the SQMS and the barriers to improvement.
6. Establish a software quality and engineering capability at AWE plc by designing a training methodology from which trainees apply knowledge and skills to improve business practices that demonstrate a cost-benefit and a

return-on-investment. This training should be implemented and tested at AWE plc.

7. Design a cohesive and congruent software quality management system having collated detailed concerns and problems from practitioners and facilitated an extensive consultation process. The design should include a logical process flow and visualisation techniques to communicate the top-level requirements.

8. Test the general applicability and value of the research findings by:
   - publishing in peer reviewed academic conference proceedings and journals
   - tracking the level of improvements at AWE plc and the resulting cost-benefit and return-on-investment
   - facilitating its wider use in industry by companies and organisations in the UK and USA.

1.7 Thesis Layout

1.7.1 Chapter 2, Literature Review

This chapter captures the comprehensive literature review conducted in support of the research. The direction for literature review was driven by the aims and objectives documented in Chapter 1. The review identifies gaps in current knowledge, drivers and strengths and weaknesses on the various topics and processes covered during the research. Case studies, conference papers and web sites related to this research are interrogated and help refine both the approach undertaken and the direction for the objectives.

1.7.2 Chapter 3, Research Methodology

Discussed in this chapter are the various research philosophies and approaches available the researcher. The philosophies are reviewed and explained and the methodology adopted for this research placed into context. A description of the

research events and investigations followed to meet the research objectives outlined in Chapter 1 are conveyed with reference to the theoretical research philosophies.

1.7.3 Chapter 4, Approaches to Internal Auditing – A Case Study

This chapter is based on "Approaches to Internal Auditing – A Software Quality Assurance Case Study" (Elliott et al, 2005) in the proceedings of the Software Quality Management (SQM) conference in April 2005, and "Towards Real Process Improvement from Internal Auditing" (Elliott et al, 2006a), in the Spring 2006 issue of the Software Quality Journal.

Conveyed here is the analysis of audit findings that should identify the problems practitioners have in implementing a local software quality management system regime. Unfortunately the data itself led to the need to review some of the approaches taken when conducting an audit. Discussed are the techniques used and the motivation of auditors and also highlighted are problems with audit sampling. Recommendations to overcome these problems are made in this chapter and these were tried out in practice and are presented in Chapter 6, demonstrating how internal auditing can be a more cost-effective, problem-solving, management tool.

1.7.4 Chapter 5, An Improved Process Model for Internal Auditing

This chapter is based on "An Improved Process Model for Internal Auditing" (Elliott et al, 2007d) published in the Managerial Auditing Journal, June 2007.

This chapter describes an investigation into the reasons why internal auditing is often perceived to not add value. A detailed analysis of the auditing process is conducted to identify areas of potential inefficiency and conflict. This chapter then recommends a new process model and approach that will improve the actual and perceived value of auditing. A questionnaire survey was sent to auditors and auditees at AWE plc, to gain their views on audit effectiveness and quantify their perception of value. The results show variations in the perceived value of internal auditing, particularly of some key stages in the process. A conclusion was that the management of internal

auditing is too focused on programme achievement, not the resulting value from improvement action. The recommended improved process model and Cost-Benefit Audit Methodology (C-BAM) approach was found to be unique within the scope of the literature review.

1.7.5 Chapter 6, An Analysis of Software Quality Management at AWE plc.

This chapter is based on "An Analysis of Software Quality Management at AWE plc" (Elliott et al, 2006b) in the proceedings of the Software Quality Management (SQM) conference in April 2006, and in Volume 15 of the Software Quality Journal (Elliott et al, 2007a).

Presented here is a detailed question set that was used to test and measure the true extent that the software quality management system was adopted and implemented across AWE plc. The results obtained from 55 implementation assessments on software systems are collated into management groups and the associated cultures discussed. The data analysis revealed the specific topics of weakness that at times reflect the cultural acceptance or resistance that management groups have towards the adoption of quality systems. From the knowledge gained on the detailed problems and barriers, effective strategies and programmes are suggested to improve the level of implementation and, therefore, the effectiveness of a software quality management system.

1.7.6 Chapter 7, Effective Facilitated Self-Assessment

Presented in this chapter is the tool that was at the foundation of measuring and facilitating actions needed to improve Software Quality Management at AWE plc. How the tool was used, and some of the successes and failures in its use are discussed. A survey was used to tease-out some of the reasons for the reticence to its utilisation. A number of strategies for its improved use are presented from this analysis.

1.7.7 Chapter 8, Providing Demonstrable Return-on-Investment for Organisational Learning and Training

This chapter is based on "An Providing Demonstrable Return-on-Investment for Organisational Learning and Training" (Elliott et al, 2007c) in the proceeding of the International Conference of Software Process Improvement Research into Education (InSPIRE) in April 2007.

In establishing a software quality and engineering capability at AWE, a holistic approach to training was undertaken. A best practice evaluation framework was used to demonstrate cost savings relating to improved effectiveness and efficiencies that were aligned to the business objective of improving the understanding of software quality and engineering concepts and its subsequent implementation. Return-On-Investment examples are also discussed. The evaluation framework considers trainee satisfaction and the capturing of improved knowledge and skills, the extent this knowledge has been applied with subsequent efficiency savings.

1.7.8 Chapter 10, An Evolutionary Cultural-Change Approach to Successful Software Process Improvement

This chapter is based on "An Evolutionary Cultural-Change Approach to Successful Software Process Improvement" (Elliott et al, 2007b) in the proceedings of the Software Quality Management (SQM) conference in April 2007.

This chapter presents the success of the methodology in terms of being an evolutionary and cost-effective means to achieve implementation of the software quality management system. The success of the methodology is presented in three ways: firstly by direct measurement, demonstrating the improved level in implementation, secondly, by an examination of the reduction in resources needed to achieve this improved level as compared with that used previously, and lastly, by the cultural change that took place, highlighted by the differences in behaviour before and after, or indeed during the achievement.

1.7.9 Chapter 10, The Value of Software Quality Management System Implementation

This chapter discusses further the use and value of self-assessment and why and it may not always be undertaken. A survey questionnaire is used in a face-to-face environment to identify the barriers to its use. Also reviewed and discussed are some cost models associated with the software quality assurance process. A follow-up survey on the perception of value to internal auditing is also presented to successfully test the quality of the improvement suggestions highlighted in Chapter 4.

1.7.10 Chapter 11, Summary, Conclusions and Recommendations for Further Work

This final chapter concludes the thesis, referring back how successful the aim and objectives outlined in Sections 1.5 and 1.6 were met. The benefits of the Software Quality System methodology developed in this research are expressed and how it can be utilised by any large or small organisation is considered. Suggestions for further work are also conveyed.

1.8 Summary

The research imperative for this thesis stems from the need, by AWE plc, to manage software quality in an effective and efficient way that provides tangible benefits and a true Return-on-Investment. With a company as large as AWE, this presents a serious challenge, particularly with the diversity of cultures that bring with it varying degrees of acceptance to new ways of working. As the world's reliance on software continues, this successful methodology will provide extensive benefits.

Chapter 2. Literature Review

Chapter Preface

This chapter presents an analysis of existing literature on software quality management and the range of methods, tools, techniques and cultural issues that support or impact its adoption and subsequent implementation. The first few sections of the chapter investigate literature associated with the internal auditing process, with a focus on auditor motivation and the drivers that have influenced current practice. The differing software quality standards are then critiqued with a useful illustration of why they are developed. The impact of quality initiatives have on organisation culture is then discussed and its importance highlighted. The significance of the role of training to any improvement activity is then conveyed together with an effective evaluation framework to help companies ensure training and learning is being correctly applied. The successes of the use of self-assessment as a supporting tool is analysed. The chapter concludes with a report on the risks associated with software projects and the models available to calculate cost saving and return-on-investment in applying quality practices.

2.1 Introduction

This literature review introduces the reader to issues surrounding the audit function. There are two chapters, 4 and 5 on the internal (to a company) audit process that also discuss the wider application of auditing in general and how external factors such as standards and guidance have influenced internal quality auditing and its development.

2.2 Background to Auditing

One of the major factors to investigate within this research is why auditing is not a popular process to go through (Cangemi, 2003). Audits are just not well received, yet are a very frequent activity. Culturally, they could be seen as a necessary evil.

According to Wealleans (2000), they are often viewed as a means to finding and recording of trivial issues, and this has blighted the perception of the auditing process. In the "audit world" this is very clearly recognised, many publications focus on this issue. Indeed a common term "added value" (Hutchins, 2002) is prevalent. Yet despite this understanding, the negative perception of the process has not significantly improved, suggesting the new focus may not have been effective.

2.2.1 Audit Definition

So what is auditing? If we take the Oxford dictionary definition (OUP, 1996), "An official examination of accounts," we find a financial bias. Indeed the populist view of auditing would be the checking of a company's financial accounts to demonstrate honesty and accuracy in its business dealings. Russell (2001) points out that this would enhance the standing and credibility of the company and probably attract and retain customers. The enhancement of credibility was picked-up by the quality profession in the 1940s (Arter, 2000) and first applied to the military and then, in the 1960s, the nuclear power industry. Arter (1989) defines four main types of auditing, financial, product, process and system. This research is focussed on the quality management aspects of auditing that provide a similar credibility role, in checking the managerial arrangements are compliant with a declared standard, specifically, ISO 9001:2000. The certification process, if conducted by an independent company, is referred to as a 3rd Party Audit. If certification is conducted by a customer, it would be known as a 2nd party audit. So internal auditing conducted by employees is often referred to as 1st party auditing. The research set out to collate and understand the problems from this internal, 1st party auditing, at AWE plc, in relation to the implementation of a defined company quality management system.

2.2.2 Audit History

The history of management system auditing is explicitly likened to the history of quality systems themselves. Pronovost (2004) reports that, in the UK, the MoD placed inspectors in factories as a means to stop bombs going off. At this time munitions suppliers had to write down the procedures for making their product, then

to make sure that workers carried-out those procedures by inspecting their work, and finally the whole method of working was inspected by government inspectors. Quality, in its various forms, went through a series of initiatives described by Pronovost (2004) until the British Standards Institute (BSI) published BS 5750 in 1979. A key element of that standard was the need to undertake internal quality audits, and the wording is very similar to the modern ISO 9001:2000 clause 8.2.2.

2.2.3 Audit Standards and Drivers

Companies seeking certification to the ISO Standard 9001:2000 derive their internal audit schemes from the clause 8.2.2, which stipulates the need (or mandatory requirement) for an internal audit process. The clause requires companies to conduct audits at planned intervals to ensure the quality management system complies with the standard and is effectively implemented and maintained. Topics or areas of work considered important by the company should have a priority over more mundane activities. Information from previous audits should to be considered. The output of this planning activity would be an audit programme. These internal audit functions are heavily influenced by the ISO Guide 19011. This guide is applicable to organisations needing to conduct internal or external audits and describes how to manage an audit programme. A key principle described by this document is that of independence, as the basis for the impartiality of an audit. This impartiality is a key theme within the ISO/IEC guide 62, which provides guidance to certification organisations. Certification bodies are prohibited from providing services or consultancy. This is understandable for an independent certification organisation, that could be accused of gaining from an assessment and thus contravene the impartiality principle. Unfortunately this concept is also prevalent in the internal auditing process, yet there is not really a need. Indeed the provision of guidance would add to the value of internal auditing. It would therefore appear that the guidance and drivers influencing certification activities have adversely affected the way in which internal audits are conducted.

2.2.4 Auditing Techniques

There are approaches and techniques that can be used in internal audits. The inappropriate use of sampling techniques is highlighted by Arter (1989) as is the need to use checklists to ensure a holistic approach is taken. Arter (1989) suggests that the decisions on sample technique should form part of the audit planning process. Cangemi (2003) agrees that many of the techniques to be applied should be decided at the planning stage. Sampling can be probabilistic, or non-probabilistic, judgmental, blocked or statistical (DoE, 2003). Arter (1989) also suggests that sampling should be based on just a percentage element and that too much time can be spent on the science of sampling.

The use of an audit checklist can be summarised (ISO/IAF, 2003) by its advantages as an aid to planning, ensuring a consistent approach and a means to collect evidence. It can also save time if presented in advance, although such a list can be intimidating and may limit focus or be restrictive or hinder communication. The use of standardised or generic checklists available from many commercial companies is not considered good practice (Arter, 1989). According to Davidson (2004), audit checklists have a role to play in measuring the effectiveness of a quality management system if they can be utilised as a self-assessment tool, the implementation rating system presented in Chapter 6 being an example of this.

2.2.5 Auditing - Summary

In the development of this literature review the aims and objectives of this research can be re-assessed. It is clear from the numerous publications that the auditing world is aware of the criticisms of the auditing process. The term "added value" is prolific within these publications, which is recognition that there has been a problem. Unfortunately, there remains a common theme that auditing will add value by finding non-conformance to a standard means of working. It is an assertion within this research investigation that a tangible cost benefit has to be attributed to the corrected action, otherwise this activity will not necessarily actually add value. The non-

conformance may have, in fact, been an effective means of working where, say, the cost of evidence, or paperwork, may have negatively impacted on progress or quality.

It is clear from the literature that improved methods of auditing would make a valuable contribution to effective working practices, if improved methods of auditing can be developed. It is, therefore, necessary to explore flaws in current auditing systems, whilst retaining the effective elements. This will take a structured approach to clearly identify these issues. The ISO 9001:2000 standard and its subsequent variations will be around for a long time, so will the derived auditing systems. So it can be seen that demonstrated improvements in auditing systems will be extremely worthwhile.

2.3 Software Quality Management Systems - Standards and Culture

2.3.1 Introduction

This section of the literature review covers two main themes: firstly, the problems organisations have with implementing their software quality systems on their software projects, and secondly, the key organisational cultural issues that exist that present a barrier to that implementation. Quality standards are developed from known best practices i.e. activities, methods and tools that have proved to be successful on projects. These activities are then grouped together into a systematic, normally generic, framework as a "standard" framework. This approach has been captured well, and included in this literature review, in the case study undertaken at IBM by Radice et al (1985) as a programming process study. Humphrey (1989) describes how this was a significant development in the world of software quality management standards as the maturity model formed the basis of the Software Engineering Institute's, Capability Maturity Model (SEI CMM) some years later.

2.3.2 Quality and Software Related Standards

Many standards exist to provide a framework for a software quality management system defining the organisational arrangements and the working processes. The best known are the ISO 9001 standard (ISO, 2000) and the Software Engineering Institute's (SEI) Capability Maturity Model Integration (CMMI) (SEI, 2001). The guidance (ISO 9001, 2004) to the revision of the ISO 9001 standard from 1994 to 2000 highlights eight principles for a successful management system: customer focus, leadership, involvement of people, process approach, system approach to management, continual improvement, factual approach to decision making and mutually beneficial supplier relationships. As the SEI CMMI conveys five levels of maturity for a company to be world class, the initial key process areas to obtain the first level of maturity would be the most important. These consist of: configuration management, quality assurance, sub-contract management, project tracking and oversight, project planning, and requirements management. There is a correlation between these two standards, various management processes and requirements management, but there are also some differences, suggesting disagreement on what would be a magic formula for a software quality management system. Further correlation of the importance of management processes and requirement management comes from the Standish Research Group's Chaos Report (Standish, 1994), which stated that the top eight reasons why projects are cancelled or severely fail to deliver on time, cost or performance, were: incomplete requirements, user involvement, lack of resources, unrealistic expectations, lack of senior management support, changing requirements, inadequate planning, system no longer needed. Again, five relate to requirements management, and three to what would be considered general management processes. The follow-up Standish Report (Standish, 2001), highlighted its top ten factors for project success as: executive support, user involvement, an experienced (project) manager, clear business objectives, minimized scope, standard infrastructure, firm basis requirements, formal methodology, reliable estimates, others, including: small milestones, proper planning, and competent staff and ownership. Again there is some correlation, but there are also differences.

2.3.3 A Programming Process Study

Radice et al. (1985) reports that a programming Site Study group was formed to review the work of eight large system programming development locations within IBM. The aim was to evaluate them according to a set of process stages. Radice et al. claim that the lessons from the study would be transferable to any project in the software industry. It was also felt that it was a means to improve the consistency and control of future projects. This provides a local example of how quality standards are formed and presents an effective methodology for any software process improvement strategy. Specifically the aims were:

- understand how the software is being produced
- propagate better best practice across sites
- help the sites in the evolution of a consistently repeatable discipline for developing software

At each site, both the managers and the programmers were interviewed to determine what processes were actually being performed against what was defined. These data were gathered with the use of a process stage versus attribute matrix as shown in Table 2.1. The process stages were: requirements, product design, component design, module design, code, unit test, functional verification test, product test, system test, package and release, early support program and general availability.

The attributes were:

- Process – the systematic flow and relationship of tasks and information needed to produce a product.
- Methodologies - the systematic procedures and techniques used to accomplish a task.
- Adherence to practice – the consistency of following defined procedures
- Tools – the automatic support of tasks and methods
- Change control – the methodology by which all changes to the product are controlled

- Data gathering – to illustrate process performance
- Communication and use of data – the effective analysis and communication of data to improve the process
- Goal setting – targets set to improve the process
- Quality focus – the pursuit and achievement of quality improvement
- Customer focus – the achievement of customer satisfaction, whether the end user or next person or group in the process
- Technical awareness – knowledge of state of the art products and processes

The team that undertook the survey were selected on their technical knowledge and ability but also their extensive training in interviewing techniques. Instructions were given to interviewees particularly in their preparation. They had to present for 10-15 minutes and talk on their process, covering inputs/outputs and any sub-processes, highlighting their responsibility. The specific questions to answer included:

- What documents do you use/produce?
- What tools do you use?
- Do you have defined procedures?
- Do you follow them?
- How do you verify quality, accuracy, and completeness?
- What checks and balances, reviews and approvals are involved?
- How do you track progress?
- How do you estimate resources needed (schedules and man power)?
- How are changes controlled?

Other questions included:

- How are the necessary skills developed/maintained within your function?
- What is the level of customer awareness within your function?
- What is your function doing to improve the quality of its process?
- What are the process problems associated with doing your job?
- What can you think of that would most improve your process?

- What can you think of that would most improve your process activity (tools, procedures etc)?
- Was the process used in this release different from that of the previous release?
- What changes, if any, do you intend making to the process for the next release?

Table 2.1. The Radice (1985) Process against a Maturity Grid

	1. Traditional	**2. Awareness**	**3. Knowledge**	**5. Skill and wisdom**	**5. Integrated management system**
Process	Not defined or used	Defined but variable	Defined but static	Defined and improving	Leading edge, integrated into the business
Methodologies	Not aware of state of the art	Aware but no use	Aware but casual use	More use than not	Fully exploits state of the art
Adherence to practices	None to little	Some but not consistent	More consistent	Consistent	Consistent across process
Tools	Not aware of state of the art	Aware but no use	Aware but casual use	More use than not	Fully exploits state of the art
Change control	None to little	Inconsistent, lacks enforcement	Follows local process	Follows process but is compromised	Follows process, is not compromised
Data gathering	None to minimal	Inconsistent	Active across parts of process	Active across process, state of the art	Completely integrated and evolving
Communication and use of data	None to little	Some but not consistent	Consistently using feedback	Pursuing improvement through information	Fully integrated as way of doing business
Goal setting	None to little	Beginning :no feedback	Established: little feedback	Active at all levels: some feedback	Active and evolving with full feedback
Quality focus	None to little	Some but not consistent	Knows how to improve	Actively pursuing improvement	Fully integrated as a way of doing business
Customer focus	None	Some focus: no feedback	Some focus : some feedback	Major focus with some feedback	Major & integrated
Technical awareness	Minimal	Aware of opportunities: minimal use	Pursuing opportunities	Vigorously pursuing opportunities	Fully pursuing professional & technical opportunities

2.3.4 Quality Culture

Organisational culture does not feature highly in these standards or reports, although, as Spencer et al. (1992) state, competency normally encompasses motivation along with knowledge and experience. Spencer (1992), defines culture as an invisible, arbitrary system of symbols with meaning, including language and ways of speaking, tools and ways of using them and ways of influencing and being influenced. Culture is also conservative about change. One thing that culture preserves is its products, so they can be studied to learn about the processes that produced them in much the same way that archaeologists study the levels of technology from the remains they dig up from ruins (Wagner, 1999). Wagner (1999) also suggests that motivation can be influenced by cultural factors. Indeed many exceptional technical books on software quality management do not focus on culture, only occasionally referencing the issue (e.g. Somerville and Pressman).

Horch (1996) warns of the pitfalls: starting a software quality programme is doomed to failure with inadequate preparation, misused terms, lack of planning and failure to recognise the individual role of members of the organisation. This emphasises, yet again, the management issues and the involvement of staff. Galin (2004) suggests that having colleagues involved in the development and, at least, the review of company procedures, will help convince other colleagues to abide by them. The little attention to culture is perhaps surprising, indicating that the effect of culture on the adoption and implementation of a software quality management system has been under-estimated (Siaksa et al, 2002). Saiksa et al. (2002) report that this situation may be further complicated in a multi-national company. Parzinger (2001) agrees, stating that the style of management also has a significant impact on culture

2.3.4 Cultural Maturity

Despite this lack of emphasis on cultural issues for quality, management adoption and recognition of its impact has been researched and made available. Crosby (1979) was an early pioneer who noticed coherent patterns of culture impacting on technology processes. If he knew the quality of the product he could make predictions about

what practices, attitudes and understanding he would find inside the organisation. Radice et al (1985) adapted Crosby's "stratification by quality" scheme to software development. Later Humphrey (1989), of the SEI, extended their work and identified five levels of process maturity through which a software development organisation might grow. A human resource maturity model was then proposed by Curtis (1990). These models represent points of view of the same phenomenon. Crosby named his patterns on management attitude, Humphrey, by type of process, and Curtis, on the treatment of people. An interesting concept of cultural definitions, developed by Weinberg (1993) has been collated with the work of Crosby, Curtis, and Humphrey to indicate both cultural and process improvement or maturity as conveyed in Table 2.2. However, Weinberg warns that maturity may not be the right word, as any software culture can be successful. Indeed, moving from one cultural pattern to another, as Weinberg prefers, may not be right for certain organisations.

Table 2.2. Weinberg's (1993) combined cultural and process improvement maturity scale.

No	Culture type	Crosby	Humphrey	Curtis
5.	Congruent	Certainty	Optimizing	Optimized
4.	Anticipating	Wisdom	Managed	Institutionalized
3.	Steering	Enlightenment	Defined	Tailored
2.	Routine	Awakening	Repeatable	Managed
1.	Variable	Uncertainty	Initial	Herded
0.	Oblivious	None specified	None specified	None specified

In an "oblivious" culture, there is not a separate software development organisation, the developer is often the user and they consider themselves to be solving problems and are not aware they are performing something called a software development process. In this pattern, people could feel that they, and only they, could give themselves what they want or are able to understand what is wanted.

The "variable pattern" is the first to distinguish between a developer and user. Often management will be unaware of the benefits of quality and that is when blaming becomes an all too frequent occurrence (Weinberg, 1993). This is the pattern of the

software super hero. Weinberg (1993) states that if a group is successful it is because they have software superheroes. Weinberg (1993) studies show that this pattern can often be found in new companies where relatively young programmers are producing innovative and new products. When the company has established itself, the heroic feats of these young programmers, lives on in often-recanted myths. This pattern can occur in large companies, but more likely in specialist teams, attached to a specific product or service, whether engineering or financial. These groups of people may have procedures that do not cover the entirety of work and are often not used in times of crisis. The variations of schedule and cost will be dependent on the individual efforts and skills of the superheroes and does not reach a rudimentary predictability of level 2 of the CMM scale.

The "routine" pattern can arise from a dissatisfaction of the variations at the lower level. Software development may need large teams that require a degree of co-ordination. In this pattern a company will begin to recognise quality management but not embrace it. Knowledge of what can be accomplished is missing, how to change is then not considered, and a belief that programmers are manual production workers prevails. Procedures are put in place to control programmers and are followed "routinely" without too much understanding of the benefit. The "routine" pattern is stable when repeatable levels of statistical control are realised by rigorous project management controls. Weinberg (1993) still notes that in a crisis, procedures are not adhered to.

The "steering" pattern Weinberg (1993) states, is created through understanding not luck. Managers in this pattern tend to be more experienced and have been successful programmers. Armed with this knowledge and experience, managers tend to steer their teams as they intervene before projects get into trouble. Although procedures may not be complete they are followed and understood, even in a crisis.

Weinberg concludes that when managers from the "steering" pattern, rise in the company organisation, the activities in the "anticipating" pattern can be observed.

Process measurement and analysis will be in place to help "anticipate" potential problems. Quality improvements in projects will be significant. The final level of the "congruent" pattern is attained when the whole company is involved in improving and optimising its processes

Weinberg's cultural types can be summarised as:

Oblivious: "We don't know that we are performing a process."
Variable: "We do whatever we feel like at the moment."
Routine: "We follow our routines (except when we panic)."
Steering: "We choose among our routines by the results they produce."
Anticipating: "We establish routines based on our past experience with them."
Congruent: "Everyone is involved in improving everything all the time.

Another interesting cultural study conducted by Weinberg (1993) was one in which he gave the CMM self-assessment questions to a range of personnel at different levels in a company hierarchy which included; a vice president, 4 directors, 11 managers and 11 engineers.

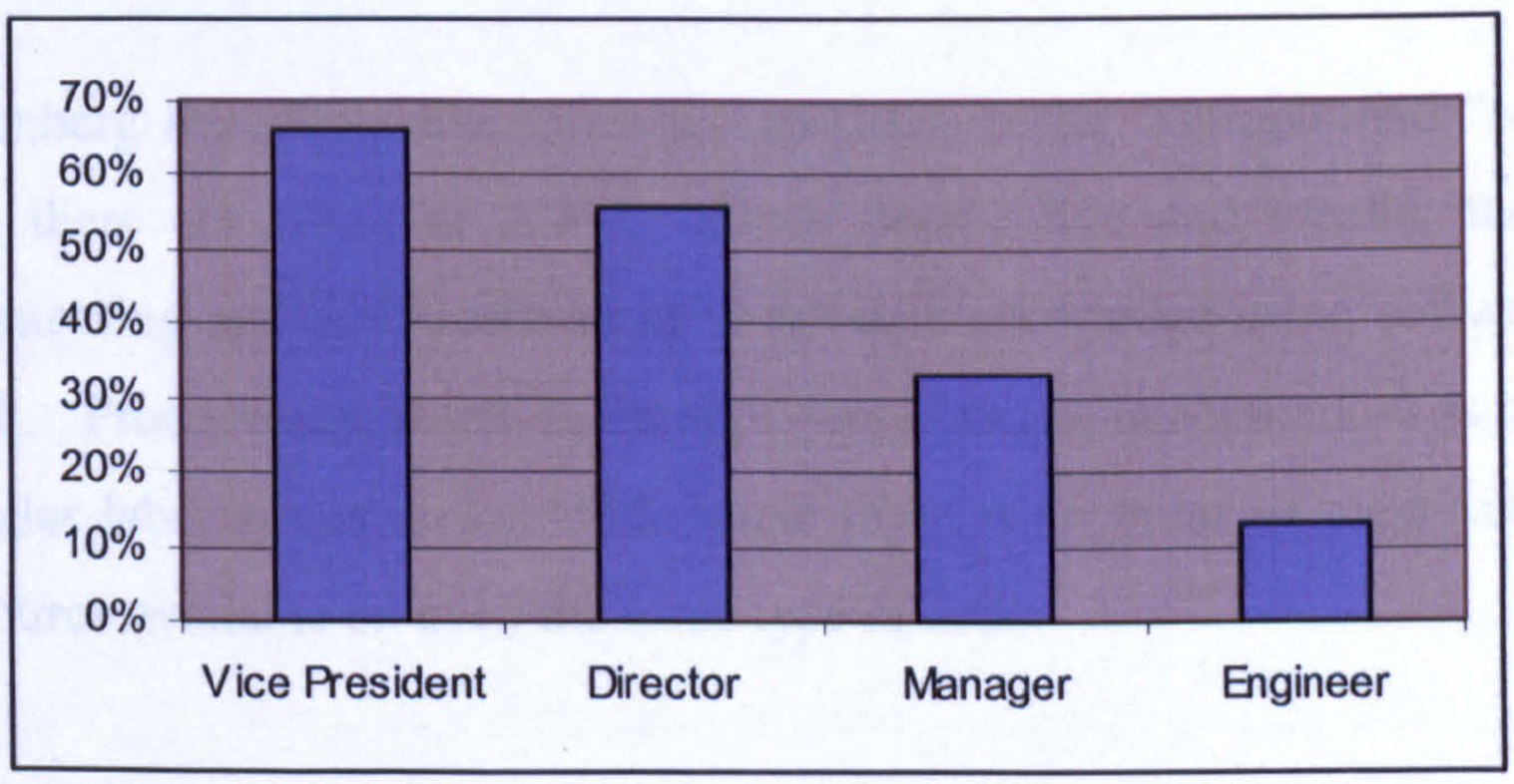

Figure 2.1. The positive answers to CMM practices provided by personnel at different levels in a company hierarchy (from Weinberg, 1993).

It can be seen from Figure 2.1 that senior managers have a more optimistic view of software engineering procedures than the engineers who actually carry them out. The conclusion from this case study sets an interesting challenge for any process

improvement initiative, as at the engineering level there is a clear need to improve processes, yet at the senior level there is not a problem.

2.3.5 Comparison of Weinberg's work on culture to that at AWE plc.

The interest of Weinberg's work on culture is that it closely aligns to the experiences the author has had at various locations and at various levels within AWE plc. Weinberg's conclusion that any cultural pattern can be successful and that maturity may not be a good descriptor for process improvement has been found to be true at AWE. There are many world class software development activities occurring at the "lower" maturity level of "variable" and "routine" patterns. So the concept that a standard way of working is successful may not always to true.

There are examples of the "oblivious" pattern, particularly in the research area. Scientists view the software as a tool in which to further their knowledge in science. One quote received in an email, tried to distinguish their work from the software quality system, stating what they did was not software development. Clearly these people were oblivious to the fact that it was..

Weinberg introduces the software superhero in the "variable and "routine" patterns and there are many at AWE. These people live and breathe their science and engineering and their motivation to advance knowledge using software is extremely high. Productivity, if that is the right word, can be benchmarked as comparable with similar laboratories in the USA where there is an order of magnitude difference in resource available on even the same type of code.

The author has witnessed the gap emphasised in the "routine" pattern between management and software developers when considering the skills as a production type activity, describing it as black art, as they can not "see" the production process. Further, it is a common occurrence that when a crisis hits the management hierarchy, all procedures "go out of the window".

For local management groups at AWE that resemble the "steering" pattern, they are using their results and data in which to make decisions on effective and efficient practices. These groups are indeed led by the software superhero that has an additional talent as a manger. These tend to know the pitfalls and do act in a quality assurance role, becoming activity engaged in the review of software development activities and the sign-off guarantee of quality on outputs from the various stages. The challenge for AWE is to reach the "congruent" pattern where the whole company is involved in quality improvement.

Weinberg's cultural study, using a questionnaire on the CMM activities known at various levels in a company's hierarchy as depicted in Figure 2.1, resembles the research activity in this thesis when baselining the implementation of the software quality system. It was clear that there was an assumption on the part of senior management at AWE that the quality system was well implemented and it was quite a shock when the level was published, albeit to a limited audience. This also influenced the strategy going forward, yet interestingly, these data was used to justify the continuing role of a central quality manager when a significant reduction in central quality roles within a central assurance group was proposed. In this respect, these data saved the author's job!

2.3.6 Summary - Standards and Culture

This section of the literature review has highlighted the importance that culture plays in any process improvement initiative. It is perhaps then surprising that overcoming cultural resistance is not included in a standard quality framework. It is left to the marketing elements of a standard, pamphlets, training courses, books etc., to highlight its importance. Key to any successful change management programme is educating the organisation on the benefits that the implementation of best practices has and, in particular, how it will make the work of an individual that bit better.

2.4 Training

2.4.1 Introduction

According to Sandi et al. (1996), training programmes are an essential feature of organisational life. Training initiatives are widely acknowledged to be a salient feature of the competitive organisation's corporate strategy and, in times of great change, learning is the key skill (Tennant et al, 2002). Employees, managers and organisations rely on training as a solution to enable issues to be resolved, yet Hale (2003) reports that only 35% of UK companies have measured the effectiveness of their training and development programmes. To value and reward learning, an organisation must have a method of determining performance in learning and gaining knowledge. Morey and Frangioso (1998) state that a method to assess the value of learning is to measure the performance of employees in creating items of business value from the learning. They suggest that for training to be effective it must have specific objectives and outcomes which directly lead to business benefit. It is surprising that despite heavy investment in training, organisations fail to evaluate adequately the value or success of their training programmes and many that do, do so inadequately (Hale, 2003).

2.4.2 Training Evaluation

A method to assess the value or impact of learning is to compare the activities and outputs against a recognised industry best practice model. One such model is the Kirkpatrick (1998) training evaluation model, also reported by Turner (2006). This simple, yet effective evaluation system presents four levels of attainment as can be seen in Table 2.3. At Level 1 success of the training is assessed in terms of satisfaction and planned action. Level 2 requires the measurement of improved skills or knowledge. Level 3 captures the application of skills and knowledge "back-on-the-job". Level 4 measures the resulting changes or improvements in the business from this applied knowledge.

Philips (2002) extended the Kirkpatrick model to include the concept of Return-On-Investment (ROI). Here the business impacts are converted to a monetary value and compared to the cost of investment in the training and all other contributory factors. Philips (2002) shows that the most common form of training evaluation consists of trainee reaction. He warns that delegates stating they are satisfied in attending a course, does not necessarily mean that their new knowledge or skills will be applied. A criticism of Kirkpatrick's model is that there is no initial evaluation to put the training into context, or identify the business need, and although there is the potential to realise that resources have been wasted, this comes later in the process. Another model to consider is Context, Inputs, Reactions, and Outcomes (CIRO) described by Morey and Frangioso (1998). This deals, in part, with Kirkpatrick's weaknesses, but lacks the assessment of business impact. Turner (2006) describes five organisational levels where learning outcomes are achieved which introduce personal motivation as a drive for learning. The five levels conveyed are, business, organisation, team, professional and personal.

Table 2.3 Kirkpatrick's training evaluation frameworks (Kirkpatrick, 1998).

Level No	Level Title	Description
1	Reaction, satisfaction & planned action	Measures participants' reaction to and satisfaction with the learning experience and captures planned actions.
2	Learning	Measures the learning and improvement in knowledge and skills.
3	Application & implementation	Measures changes in on-the-job behaviour and progress with planned actions.
4	Business impact	Measures the changes in business impact. Has the training helped towards the operational objectives?

A survey by Tennant et al. (2002) of US companies evaluating training programmes found 13% did not carry-out evaluations, 52% used satisfaction as the key evaluation technique, 3% used skill acquisition, 17% assess impact on the job, and 13% gathered data on changes in organisational performance. Tennant et al (2002) also suggest that

another benefit gained by evaluations of training programmes is that they can be used as a diagnostic technique to improve the training or they can help decide on alternative training.

Dwyer (2001) asserts that the effectiveness of the training itself can be improved by applying modern thinking on learning styles and treating course attendees as customers with individual needs. In this respect his new training model is learner-centred which considers preferred learning styles (Kolb, 1984) taking into account how the brain works and the environmental conditions for learning described by Morey & Frangioso (1998) which include the emotional, physical, and social. It extends the usual context-centred model where the main focus is on the cognitive environment. Dwyer's (2001) research on how the brain works concludes that the ideal learning environment has the qualities of:

- being emotionally safe
- being free from intimidation and rejection
- being high in acceptable challenge
- having active participation and
- being a place where learners can experience a relaxed alertness.

According to Dyer (2001), food plays a part as a nutritious breakfast containing high levels of protein will provide alertness during mornings and then a balanced diet containing carbohydrates will maintain alertness throughout the day. Caffeine drinks act as diuretics and deplete the body of water the brain needs to learn. Dyer (2001) goes on to say that course design usually considers the need for more active learning in the afternoon than the morning, but consideration should be given on the fluctuations of neurotransmitters which occur every 90 minutes. New concepts should be introduced in the first 20 minutes followed by 10 minutes of "downtime" where learners process the new information and are able to make meaning of the new information. The next 10 minutes should be used to further elaborate, reinforce, and summarise (Dyer, 2001). Kolb's ideas (1984) on learning styles have been used to

introduce new ways of teaching which embrace the learner-centred focus. A questionnaire helps to identify whether someone is an Activist, Reflector, Theorist, or Pragmatist. Knowing your learning style can help participation in more suitable learning opportunities. Kolb (1984) also describes a concept, where it is believed that experience plays a central role in the learning process. He proposes a four-stage cycle for learning:

1. the concrete experience,
2. observations and reflections,
3. formation of abstract concepts and generalisations,
4. testing implications of concepts in new situations.

2.4.4 Summary – Training

The ultimate aim of the AWE software engineering training was to improve the application of the software quality management system. Training and education are a key component in any improvement initiative. To further evaluate the success of this training in terms of value, the financial benefits should be considered as outlined by Philips (2002).

2.5 Self-Assessment

2.5.1 Methods of Self-Assessment

Lee and Quazi (2001) report that the recent increase in the use of self-assessment as a process improvement tool has been, in part, due to organisations setting the goal of attaining a recognised quality award. These awards include The Malcolm Balridge National Quality Award (MBNQA), and The European Quality Award (EQA). An integral part of obtaining such an award is to conduct self-assessment of organisational performance against the award criteria or framework. Lee and Quazi's (2001) analysis of the frameworks of these quality award systems concludes that there are differences in "assessment items" or "areas to address" and their associated

weightings. However, Van der Wiele et al, (1995) note that there are sufficient similarities to state that any one would be suitable to facilitate a quality or continuous improvement initiative. A number of surveys exist that provide a good insight into how and why self-assessment has been utilised in many organisations (Samuelson and Nilson, 2002). However, there is some evidence that self-assessments are conducted purely as an internally driven initiative for the purpose of improvement, and not for external recognition (Lee and Quazi, 2001).

A key element of the self-assessment process is the question-set. Typical question-set methods are pro forma or matrix as described by Samuelson and Nilson (2002). The European Foundation for Quality Management (EFQM) pro forma approach requires the production of 32 forms for each sub-criterion in order to identify the strengths and areas for improvement for the company. The matrix approach requires a company to draw up its own matrix of performance. The matrix consists of a series of questions regarding approach and deployment on a scale ranging from no deployment, to full engagement, which are linked to each EFQM criteria. Samuelson and Nilson (2002) point out that responses to questions need to be carefully managed as issues such as bias and inconsistency can distort analysis, though inconsistency can be minimised with a suitably large number of samples to "average-out" false variations. Suitable guidance or clear criteria will also help reduce this. The assessment data should then be collated to establish the result against the model criteria, and analysed to determine appropriate actions with assigned priorities.

Samuelson and Nilson (2002) go on to state that improvement actions should then be delegated to suitable management with possible targets set and, in order to maintain their commitment, people must be informed about targets, execution, and consequences of self-assessment, and improvements in profitability and overall business performance must be visible. Measuring, monitoring and communicating the results of improvements, provides this visibility and this leads to recognition. Lee and Quazi (2001) agree that improvement can not be recognised unless it is measured.

Waina, (2001), documents characteristics of the use of self-assessment. In its simplest form it can be administered in one or two hours. In this situation the disruption is low. Unfortunately, in these situations the accuracy of results can also be low. Self-assessments by software managers and software engineers tend to have unconscious bias and can be incomplete or inaccurate, as few of us admit to our own short comings. Van der Wiele et al. (1995) point out that many such problems are difficult to detect via self-assessment. Waina, (2001), also describes how self-assessment can also be used for the basis for an intensive, independent and thorough investigation into local working practice. In these situations, the disruption is high, as is the cost, however, accuracy of results is normally high. In considering which assessment method is suitable, Samuelson and Nilson (2002) assert that an organisation needs to consider the assessment objectives and desired outputs, the accuracy of results, the cost to prepare for and conduct the assessment, and the extent of organisational disruption. They report that self-assessment provides an important cultural benefit because it encourages an ethos of continuous improvement, promotes a holistic perspective, and allows people to gain a broader understanding of the business. There is sufficient evidence to indicate self-assessment is effective as surveys of the perception of value are high and increase when subsequent assessments have taken place (Van der Wiele et al, 1995).

2.5.2 Summary – self-assessment

Self-assessment can be a useful tool to any process improvement initiative although it has been highlighted that its application needs to be carefully managed as, used incorrectly, inherent bias in the way people answer questions can provide the wrong data which can be used to make decisions.

2.6 Software Project Risks

With the significant number and high profile of software project failures, as reported by Standish (1994), a considerable amount of research has been undertaken on risk concepts to try and avoid these failures (Jones, 1994). Jones (1994) describes the "Common Colds" of software management:

- Excessive schedule pressure – cultural issues that requires therapy to deal with the sociological issues.
- Poor quality – technology problem with cultural aspect. Requires new methods in measurement, defect prevention and removal, plus cultural awareness of the true value of quality to the business.
- Inaccurate estimation – essentially a technological problem, but has a cultural aspect. Requires good estimating and planning tools. However, measurement is needed for accuracy and this is where the cultural aspect comes in.

Further work by Jones (1994) concludes that the most common risks for software projects depends on the type of application as depicted in Table 2.4. These data has been developed from a large number of projects in each application domain.

Table 2.4. Jones' (1994) most common software project risks per software application

	Information Systems	**Hardware**	**Commercial Software**
1st	Creeping requirements	Long schedules	Inadequate user documentation
2nd	Schedule	Inadequate cost estimating	Low user satisfaction
3rd	Low quality	Excessive paperwork	Excessive time to market
4th	Cost overruns	Error prone modules	Harmful competitor action
5th	Inadequate configuration	Cancelled projects	Litigation expense

	Military SW Projects	Outsourced Projects	User Developer
1st	Excessive paperwork	High maintenance costs	Non-transferable applications
2nd	Low productivity	Friction between contractor & client	Hidden errors
3rd	Long schedules	Creeping user requirements	Unmaintainable software
4th	Creeping requirements	Unanticipated acceptance criteria	Redundant applications
5th	Unused or usable software	Legal ownership	Legal ownership

Jones' (1994) philosophy is that the value of identifying the risks for each software application is to proactively plan for the appropriate risk mitigation action. Significantly these can be the quality assurance activities required from the adoption of quality standards that could be captured in a quality plan. Jones' examples of risk mitigation activities for the common risks are presented in Table 2.5.

Table 2.5. Common software project risks with suggested mitigating action.

Risk	Mitigating Action
Creeping requirements	Use of prototyping, Joint application workshops (JAD), Use of recognised methodology, measures and metrics on growth, place contracts by cost and number of function points.
Schedule overrun	Technology to improve estimation and technology to reduce schedule durations
Cost overrun	Quality and reliability estimation (tools), defect prevention, defect removal, quality measurement
High maintenance costs	Structural analysis (tools), restructuring tools, reverse engineering tools, re-engineering tools, error prone module analysis and removal
Inaccurate metrics	Use function point not lines of code

2.7 Financial Models

Most financial models for software systems have been produced for estimating the cost of software development such as the COCOMO (Constructive Cost Model) model developed by Boehm (2000). The original model was constructed from data collected from 63 projects from 1964 to 1979. These projects were quite diverse in terms of the environment in which they were implemented, the design methodologies and types of languages used. The model assigned project size on source instructions, although there was an understanding that source instructions were not uniform. A small project would typically have around two thousand lines of source instructions and a large project would be around 100,000 lines of instructions. The project types were described as organic, semi-detached and embedded. An organic project would be developed by a small team in a familiar environment that was well known to the developers and would be small to intermediate in size. An example would be aircraft flight post data analysis software. A semi-detached project would be intermediate project in size and a mixture of embedded and organic characteristics such as an aircraft simulator training system. An embedded project would be under tight time constraints and regulations, could involve formal methods and techniques that would be supported by a team of analysts and a large team of programmers. An example would be an aircraft on board collision avoidance system. Of interest was that the average projects had life cycle phases whose costs broke down to 6% for both planning and requirements management, 16% for design, 24% for detail design that included the actual coding, unit testing as 31 % and integration testing as 23%. More modern thinking by the Standish Research Group (2001) on requirements management states that the requirements phase is so important that it should account for 25% of the project time and testing can be as much as 40% of the total project costs. At the "basic" form of the COCOMO model, an estimate of the number of man-months that a particular development project will take is presented from an assessment of the number of lines of code that are likely to be produced. At the

"intermediate" level, the model uses a range of development processes, system and people attributes to refine cost and schedule estimates (Boehm, 2000).

2.7.1 Financial Effectiveness Models for Quality Practices

In considering the cost-benefits and value of software quality or engineering practices the effectiveness of reviews, particularly software inspections (Freeman and Weinberg, 1991) are often cited. For example, Freeman and Weinberg (1991) report that they cut testing costs by 50% - 80% or that they remove 80% - 95% of faults at each development stage and Gilb & Graham (1993) state that they can reduce schedules by up to 25% and produce a 28 times reduction in maintenance costs and can cost only 5% - 15% of development cost. These are supported by Yourdon (1989) who states that testing reviews or inspection can produce a 10 times reduction in faults reaching the testing phase. If ever a case could be made for the value of a quality assurance system, these few statements make it. The important of design information in a maintenance regime is given by Bennett et al (2000) in that with a complex system, 50% to 95% of the cost needed to make a change can be taken to understand the program. The need for independent testing is highlighted by Freeman and Weinberg (1991) in their research that testers find only 30% to 40% of their own faults. Although this is not a detailed review of all cost models associated with software development practices, it does leave a feeling that is in line with Brooks' (1987) comment that there is "No silver bullet".

2.7.2 Return-on-Investment

The SQMS at AWE had to translate two best-practice standards, ISO 9001 and Defence Standard 05-95, into a specific AWE system. The SQMS, documented in the Company Software Procedures, had to be a definitive interpretation of these standards so that employees did not need to refer back to the originals. As this research presents a cost-effective methodology, it is of interest to review the themes of Cost-Benefit and Return-On-Investment (ROI) in the work of David Rico (2002). ROI is the quantification of the benefits received in financial terms of any investment scheme

such as the introduction of a SQMS or software process improvement project. Waina, (2001) states that the drive to invest in these systems is to make the business more successful by producing better products, more quickly and cheaply whilst improving customer satisfaction. Significant investment is then required in process definition and documentation, training and education, as well as tools, technologies and techniques.

A powerful strategy to aid adoption of any management system is to demonstrate its cost-benefits and ROI. As the start-up costs and effort are apparent in any software process improvement or quality system initiative, a framework or method to convey benefits will surely provide an incentive to accelerate its adoption. Rico (2002), analyses a number of techniques and standards in terms of training costs, project costs, life cycle benefits, benefit-to-cost ratio and ROI. The adoption of standards has additional preparation and assessment costs. The systems assessed for ROI are Software Inspections (Fagan, 1979), Personal Software Process^{sm} (PSP^{sm}), Team Software Process^{sm} (TSW^{sm}), Software Capability Maturity Model (SW-CMM) (Humphrey, 1989), ISO 9001 (ISO, 2000), and Capability Maturity Model Integration (CMMI) (SEI, 2001). The conclusion from Rico's analysis is that due to the high start-up cost for the adoption of the standards SW-CMM, ISO 9001, and CMMI, they provide the lowest ROI. According to Rico (2002), the highest ROI was achieved by PSP, then the software inspections.

2.7.3 Summary - Software Project Risks and Financial Models

This section has highlighted the significance of risk management and the all important financial aspects associated with process improvement. Sedden (2000) states a particular challenge for quality practitioners is the risk in military projects and adherence to standards such as ISO 9001 that produce excessive paper that is rarely utilised. The work of Jones (1994) found that 400 English words could be produced for every ADA program statement in such a project. It can be concluded from these types of aspects that all process improvement initiatives should have a cost-benefit

business case to clearly demonstrate intended financial return on the time and resources invested.

2.8 Overall Summary

This chapter has presented a literature review on the various and numerous aspects of quality management and process improvement in relation to software projects. The numerous standards available to the process improvement practitioner have been conveyed. The purpose of capturing known best practices that have proved successful as standard practices has been conveyed with the interesting case study by Radice et al (1985). The tools that can support standard implementation have been scrutinised such as the audit function, self-assessment and training. The literature review has been focused on identifying the influencing factors that drive the way these tools are utilised, together with there strengths and weaknesses. The review has also captured up to date philosophies on these tools from other research. The importance of culture as both a hindrance and ally has been emphasised. The need to place process improvement on a sound financial footing as also been presented. The review has also identified significant gaps in the body of knowledge for process improvement as, although many standard approaches are encouraged, there is no one best approach. It is still down to the process improvement practitioners to decide their own strategies based on the best information available and their particular circumstances. This emphasises the need for this research, particularly in view of the increasing use, size and complexity of software in everyday use.

2.9 Review of Research Aims

The review has identified some significant research opportunities in both the general application of process improvement but also the tools and methods used to support its implementation. Evidence has been provided that the auditing world is in a state of disarray as it has identified that it is being criticised for not adding value. This presents a research opportunity, to understanding why audit is perceived to not add value and develop an improved method. This is in line with the research objectives 1 to 3 detailed in Chapter 1. Key to any improvement initiative is a system to measure where you are against a standard or system that will help understand issues with its adoption. Objective 4 will provide a chance to develop a case study on which to identify barriers to implementation. The review has presented an extensive failing in the approaches to company training programmes. With so few companies able to demonstrate the true impact of their training programmes on business goals, a demonstration of such an approach as identified by objective 5 would indeed be worthwhile. The need for a well defined quality management system has been emphasized by the quality standards section, how this is achieved will be met by objective 6. The research includes an opportunity to test the various recommendations suggested by presenting at related conferences, publishing in peer reviewed journals and conveying the lessons learned in the wider industrial community, as conveyed by objective 7.

This literature review has identified some serious inefficiencies in the methods utilised in the adoption of quality management systems that require further investigation. In this respect, the research aims and objectives are appropriate and necessary and fully justifies the research project.

Chapter 3 Methodology

Chapter Preface

This chapter discusses the various research philosophies and approaches available to the researcher. It seeks to review and explain the philosophies and place the methodology adopted for this research into context. A description of the research events and investigations followed to meet the research objectives outlined in chapter one will be conveyed with reference to the theoretical research philosophies.

3.1 Research Philosophies

A research philosophy is a belief about the way in which data about a phenomenon should be gathered, analysed and used. Two main research philosophies exist, the scientific positivist (sometimes called scientific) and the more subjective interpretivist or anti-positivist (Galliers, 1992). The positivist approach is based on the theory that observations are measurable, repeatable and that the correct analysis will lead to the correct answer, which will remain correct unless interfered with (Popper, 1959). This approach is favoured in the world of the science, mathematics, and engineering. There is debate as to whether the positivists approach is suitable for social sciences (Hirschheim, 1985). The interpretivist's philosophy is that a reality can only be fully understood through subjective interpretation and an intervention in that reality. The aim of both approaches is to turn what is believed to be known (doxology) into what is known (epistemology).

3.1.1 Positivism

At its most conceptual level, science is nothing more than the search for understanding. It uses whatever tools, techniques and approaches that would be appropriate to acquire the knowledge on a particular subject matter. Any attempt to acquire knowledge could be considered science. For some cultures, meditation or consulting an oracle are forms of inquiry. However, in the West we might consider this form of knowledge acquisition "unscientific" because it does not match our conception of science. Conventions are based on past experiences at acquiring

knowledge, they are conventions that have proved successful in the past. So for something to be considered scientific it must use an agreed set of conventions – the scientific method. It is the manifestation of positivistic conception of science/inquiry or "positive science" and has a long history of providing an accepted understanding of nature. Hirschheim (1985) proposes "five pillars" to summarise positivism:

1. the unity of the scientific method – the approach of knowledge acquisition is valid for all forms of inquiry,
2. the search for human causal relationships – the desire to find regularity and casual relationships among the elements of study,
3. the belief in empiricism – the belief that the only valid data is that which is experienced from the senses,
4. the value-free nature of science (and its process) – the understanding the science transcends all cultural and social beliefs held by scientists,
5. the logical and mathematical foundation of science – the provision of a universal language and formal basis for quantitative analysis.

3.1.2 Interpretivism

The interpretivists' view is that the positivist fail to appreciate the fundamental experience of life in favour of physical and mental regularities. The meaning of experience which defines human characteristics has been neglected. Individuals do not exist in isolation but need to be understood in the context of their cultural and social life. Studies by Burrell and Morgan (1979) found that human values intrude on the process of scientific inquiry and that man as a subject matter was a free spirit and not subject to the law in the physical sense. It was evident to the interpretivists that the scientific method could no longer be regarded as value free. The scientific observer was seen as an active force which determines the way in which scientific knowledge was obtained.

3.2 Discussion and rationale for choice of philosophy

In both organisational science and information systems research, interpretive research was the norm until the late 1970s (Vreede, 1995). More recently the positivist

tradition has been more popular (Dickson and DeSanctis, 1990). Indeed 96.8% of research in the leading US IS journals conform to this theory (Orlikowski, 1991). However, it has often been stated (e.g. Benbasat et al, 1987) that no single research methodology is intrinsically better than any other methodology with many authors suggesting a combination of research methods to improve the quality of the research (e.g. Kaplan and Duchon, 1988). Also a "house style" methodology (Galliers, 1992) can be the choice of many institutions. What becomes clear is that a methodology best suited to the problem under consideration, as well as the objectives of the researcher, should be chosen (Benbasat, 1984: Pervan, 1994).

The author's overriding concern is that the research approach adopted is relevant to the research question, as set out as such in Chapter 2. The author has chosen the interpretivist philosophy, which is appropriate for this purpose in particular, how employees of a company like AWE plc and the affected software quality practitioners adopt and adapt to the use of a software quality system and the range of improvement tools used to support implementation. The author's research is concerned with capturing information from the practices and social interactions in a company environment rather than a laboratory in a scientific approach, which leads towards to interpretivist philosophy rather than the positivist philosophy. The author's research involves an element of response to and interaction with the software quality management system which was not fully accepted at AWE plc and required a change in people's practices. Also the author plays a part in the implementation and improvement process. Some of the recommended improvements highlighted by the research investigations are actively tested and validated.

3.3 Research approach

The work of Iivari (1991) provides a summary of research approaches that have been adopted and used as a framework to undertake research experiments. Three board styles of research approach, constructive, nomothetic and idiographic, are described and are aligned as depicted in Figure 3.1

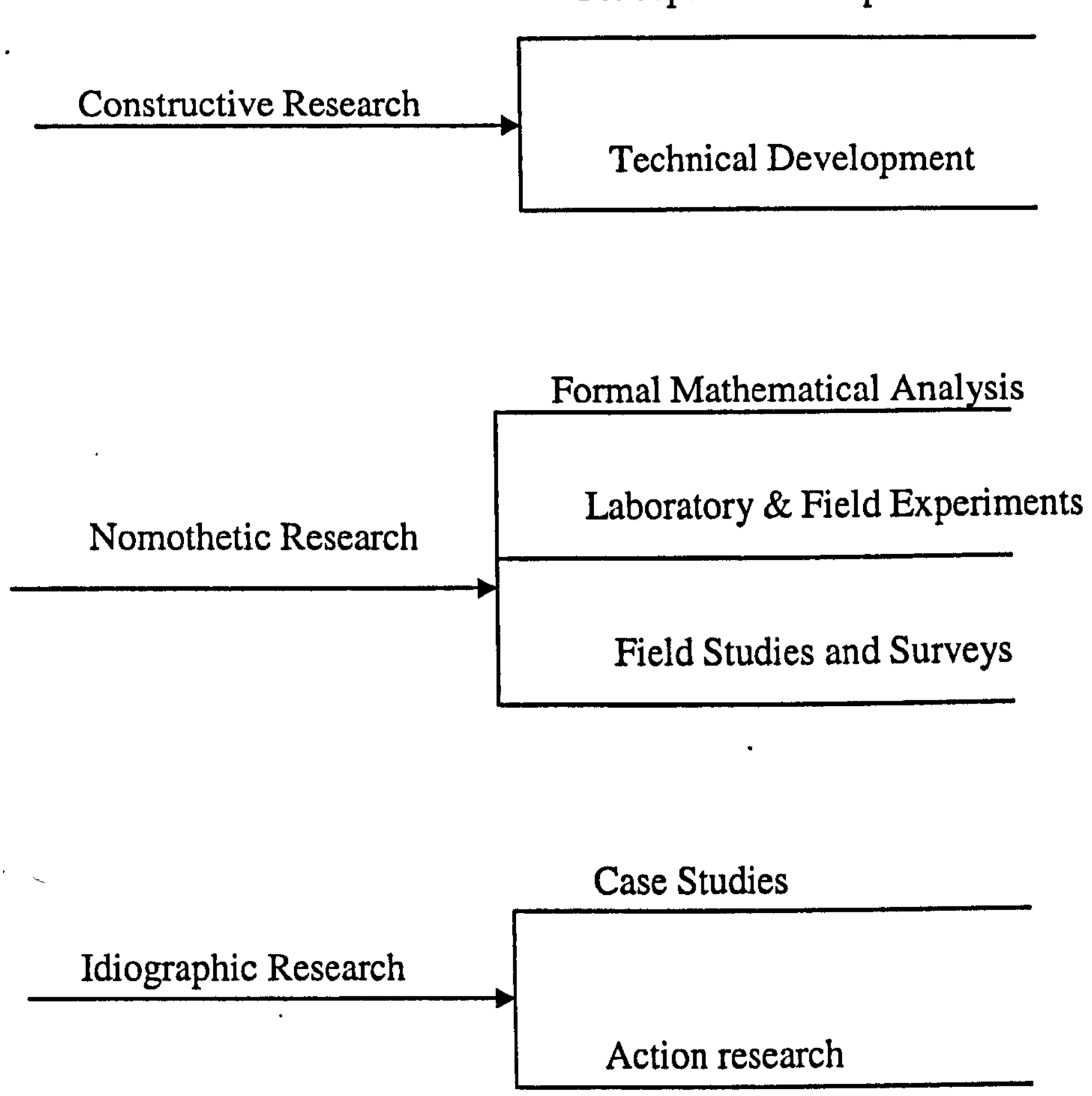

Figure 3.1: Summary of Research Approaches (Iivari, 1991)

The constructive method develops frameworks, refining concepts or pursuing technical developments. These frameworks do not describe an existing reality nor do they have any "physical" realisation (Cornford and Smithson, 1996).

Nomothetic research analyses empirical data to test hypotheses on the characteristics and behaviours of phenomena under investigation. It is the search for evidence to support general laws or theories. This method emphasises a systematic approach of hypothesis testing within the traditional science. Although initially appealing as an approach, the quest for support for general laws or theories would not be applicable to

specific organisational life as each company has different drivers to which they respond.

Idiographic research investigates events, cases or instances to accurately interpret the phenomena under investigation in its own context. This research approach emphasizes the analysis of occurrences based on participation or close association on everyday events (Cornford and Smithson, 1996). In this respect, this type of research is the essence of an interpretivist approach.

3.4 Research Strategy

Research strategy is defined as the research investigation taking on a particular style and utilising different methods with which to collate data (Galliers, 1992).

A number of research methodologies have been identified that attract the need for taxonomies. Table 3.1 (Galliers, 1992), lists types of methodologies and whether they typically conform to the positivist or interpretivist paradigm. Also indicated are the methodology used by the author in this research. The key features of each research method will be explained.

Table 3.1 A Taxonomy of Research Methodologies (from Galliers, 1992)

Scientific/Positivist		Interpretivist	
Laboratory Experiments		Subjective/Argumentative	
Field Experiments		Reviews	
Surveys	X	Action Research	X
Case Studies		Case Studies	X
Theorem Proof		Descriptive/Interpretive	
Forecasting		Futures Research	
Simulation		Role/Game Playing	

3.5 Research Methodologies

Laboratory experiments - permit the researcher to identify precise relationships between a small number of variables that are studied intensively in a laboratory situation using quantitative analytical techniques.

Field experiments – extend the laboratory experiments into real life situations or organisations, achieving greater realism.

Surveys – enable the researcher to obtain data on practices, situations or personal perceptions at a point in time through questionnaires or interviews. Conclusions can then be drawn on these data using quantitative analytical techniques. Surveys are useful for studying a number of variables in laboratory or field situations. Survey results can also be subject to qualitative analysis if open questions are used. This would then be a more interpretivist approach.

Case studies – are viable when the study takes place in its natural setting. The researcher, in trying to understand the complexity of the processes, can ask "how" and "why" questions to gain this insight (Benbasat et al, 1987). Also, there may not have been many previous studies in the area under investigation, or there is a contemporary element to the research. A feature of case studies is the collation of data from different viewpoints to see if results converge to validate any assumptions or conclusions (Yin, 1989). Data gathering can include direct observation, measurement of processes, interviews, and documentation reviews on the process.

Simulation – involves copying the behaviour of a system. Simulation is used in situations where it would be difficult to solve problems analytically, and typically involves in introduction of random variables.

Forecasting/futures research – involves the use of regression analysis and time series analysis to make predictions about likely future events. It is useful in situations of rapid change.

Subjective/argumentative research – requires the researcher to adopt a creative or speculative stance rather than act as an observer. It is a useful technique for building on new theories and testing new ideas.

Action research - is a "hands-on" approach to research. It aims to contribute to the practical concerns of people to solve problems, often in a work and organisational environment (Denscombe, 2002). The approach also aims to unearth other related issues, possibly as a result of an initial change. This leads to further observations and feedback for additional improvements.

Grounded theory - is a pragmatic approach to the analysis of qualitative data, whereby the researcher seeks to impose personal and professional qualities on the study (Glaser and Strauss, 1967). The analysis of these data should provide new theories to the topic under investigation and that these theories should be "grounded" in empirical reality, which allows for feedback and possible refinement (Denscombe, 2002). A key strategy of grounded theory is the investigation should proceed with an open mind, and should be seen through the eyes of a beginner (Glaser, 1992).

Participant observation - extends the participation element of action research. Here the researcher participates and observes regularly in the process over a reasonable length of time (Becker and Geer, 1957). It provides information on cultures and events from the insider's point of view. Care must be taken not to adversely interfere with the phenomenon under investigation.

3.6 Chosen Methodologies

It would be fair to say that the methods chosen for this research, surveys, case studies and action research are heavily influenced by the authors' position within the company as software quality manager, and the opportunities that this created. With approximately 6000 employees at AWE, the prospect of obtaining accurate results for conducting surveys is high due to the sample size. The close proximity facilitates

follow-up interviews to refine issues of debate. Proximity also proves useful in case study research as an investigation can be seen in its natural setting and the "hows" and "whys" can be tested to gain greater insight into the variables being researched. Finally, the "hands-on" approach for action research is evident as the drive for the research is to solve problems relating to proving that a good job is being done. Additional adoption and implementation of the software quality system from a baselined measure provides that proof. Solving many different problems in their natural settings demonstrates that the author is, therefore, an agent for change.

3.7 Methodology Summary

This chapter has presented a detailed account of the research philosophy, approach and strategy. The research undertaken is interpretivist utilising a mixture of survey, case study and action research approaches. However, the use of surveys presents an element of positivism too. As this research investigation is based on events within a large organisation we can see that this closely relates to the work of Yin (1989), particularly in summarising empirical studies for interpretation. The style is predominately idiographic with the frequent use of constructive frameworks in which to collate data. The final phase of the research became action oriented with improvement in working practices.

Chapter 4. Towards Real Process Improvement from Internal Auditing - A Case Study

Chapter Preface

The analysis of audit findings should prove useful in uncovering the problems practitioners have in implementing a software quality management regime. The understanding gained from this analysis could then be used to solve the issues involved, and make software management, e.g. development or procurement, more effective. This case study presents an initial analysis of audit findings that led to the need to review some of the approaches taken in gathering audit data. This review included the techniques used and the motivation of auditors. A detailed implementation rating system was devised to further investigate and accurately identify specific problems. It was also used to test and validate initial conclusions and highlight problems with audit sampling. Without proper management, particularly for the analysis of audit findings, the internal audit process can be an ineffective use of resources (Elliott et al., 2005). The recommendations made by this chapter can provide practical solutions to making internal auditing a cost-effective, problem solving, management tool.

This chapter is based on "Approaches to Internal Auditing – A Software Quality Assurance Case Study" (Elliott et al, 2005) in the proceedings of the Software Quality Management (SQM) conference in April 2005, and "Towards Real Process Improvement from Internal Auditing" (Elliott et al, 2006a), in the Spring 2006 issue of the Software Quality Journal.

4.1 Introduction

"Trust me – I'm from the auditing department and I'm here to help" the infamous first words often heard by employees on their first exposure to an audit. The auditing

process can be thought of as the face of quality as it is often the first experience people have when their working practices are tested against a company quality system or standard. If these initial interactions are negative, the feelings and perceptions formed can prove very difficult to change and in some instances, can lead to a position where recovery may never be possible.

A common driver for companies to establish an internal auditing department and company process is to comply with the ISO 9001:2000 (ISO, 2000) Standard clause 8.2.2, Internal Audit. A company seeking to, or wishing to remain certified to ISO 9001:2000 is required to conduct internal audits at planned intervals to determine whether the management systems, and its subsequent implementation, comply with the standard and is effective. The benefits for companies gaining certification could include improved efficiencies, or a means to gain a competitive advantage, or simply to improve market perception. The certification process, if conducted by an independent company, is referred to as a 3rd Party Audit. If certification is conducted by a customer, it would be known as a 2nd party audit. So internal auditing conducted by employees is often referred to as 1st party auditing. The research for this chapter, sets out to collate and understand the problems from this internal, 1st party auditing, at AWE plc, in relation to the implementation of a defined company software quality management system. The objective was to provide a clear set of actions needed to improve the effectiveness of the implemented system.

Although this objective was finally met, it was not by the method identified. Initial research questioned the means by which the audits were conducted and possible interpretations of audit findings. This required further research into auditing styles and areas of inefficiencies. Methodologies were devised to not only clearly understanding audit findings, but answer the original research question. A questionnaire was used for part of the research, but unfortunately this highlighted the care needed in setting questions. Human nature being what it is, you often get the

politically correct answer and other methods are best employed to uncover exact situations or problems. This chapter also outlines some areas for further research.

It should be stated that the subject matter for this chapter has been confined to software quality management and that the issues identified in this research may not be applicable to other management system topics.

4.2 Methodology

To initiate the investigation, 260 internal audit deficiencies raised over a five-year period for non-compliance with the company's software quality system were collated. The topics were kept as detailed as possible so that fairly precise conclusions on trends could be explored. An example was separating change control from the other elements of configuration management. To compare and verify initial conclusions, 105 2nd party audit deficiencies, raised over the same period by the customer, the Ministry of Defence (MoD), were also collated and analysed. This could be considered an extremely useful amount of data from which to draw conclusions on the precise problems practitioners have in implementing a software quality system. Unfortunately this did not prove to be the case; the information questioned both the depth of the audits undertaken and auditor's motivation to take a systematic and thorough approach.

Two distinct paths of further investigation were initiated, one to investigate the audit trail suggested by these data and secondly to devise and use a system to obtain the answer to the research question on the problems associated with implementing the software quality system. The audit trail suggested by these data was compared to a sequential life cycle for a new software project and that for a system in the maintenance phase. A questionnaire was used to test auditor motivation, which again unfortunately proved inconclusive, so follow-up interviews took place.

4.3 Software Quality Audit Findings

Internal company audit deficiencies were initially collated and grouped by topic, as shown in table 4.1. These deficiencies were raised by a number of full time and part-time auditors. The ratio of full to part time is approximately 1 to 5. The turnover of staff for part time auditors was fairly high, the significance of which was investigated. However, this is not the case for full time auditors. Over a five-year period a total 260 non-conformances were raised.

Table 4.1. List of topics and numbers of deficiencies found on internal software audit over five years.

No	Topic of Deficiency	Number of Non-conformances	%
1	Software inventory listing	64	25%
2	Software management plan	45	18%
3	Disaster recovery plan	37	15%
4	Documentation errors	20	8%
5	Assignment of responsibility	13	5%
6	Change Control	12	4.6%
7	Y2K	11	4.2%
8	Other configuration management	10	3.8%
9	Testing arrangements	9	3.4%
10	Coding standard	8	3%
11	Virus checking	6	2.3%
12	Sub-contract management	5	1.9%
13	No service level agreements	4	1.5%
14	Lack of reviews	4	1.5%
15	No project plan	3	1.2%
16	Obsolete software not withdrawn	3	1.2%
17	Security arrangements	3	1.2%
18	Requirements	2	0.8%
19	Design	1	0.4%
	Total	260	

Each local area that manages software systems, whether in-house development or acquisition, is required to appoint a person responsible for an inventory list of all the software items in that area. The inventory, which could be considered an initial configuration management activity, is required to identify; software version, location, status, person responsible and importantly the reference of each software management plan. This local Point of Contact (POC) is responsible for the approval of the local management plan that describes the local area management arrangements for its software systems. The Disaster Recovery Plan (DRP) describes these data, file and system back-up arrangements and planned recovery in the event of a system failure. The documentation errors are a combination of fairly trivial issues, such as references missing, missing/wrong approval, page numbering, etc. Assignment of responsibility relates to the aforementioned appointment and job description. The change control topic is specifically a total lack of a documented change control process when changes are being made to existing systems. The issues on Y2K include lack of assessment of threat, no date testing or the non-availability of these test records. The other configuration management topic includes lack of configuration identification and baselines, a lack of descriptions for items under a status accounting system, and not undertaking configuration management audits. There was an even distribution of testing problems including; no specifications, no plans or procedures and a lack of test records. The lack of existence of a coding standard, to standardise code production, was reported on eight occasions. The remaining topics had fewer deficiencies raised, but are important in terms of achieving software to time, cost and quality. Poor requirements management accounts for five of the top eight causes why software systems fail to deliver on quality, in terms of the intended benefits (Standish, 1994). Further research is required to investigate whether these important topics have been checked as often on software quality audits or are in fact well managed.

The audit non-compliances raised by the company's main customer on 2nd party audit were also collated in a similar manner to the internal audit data to compare findings

and seek correlation. There were 105 deficiencies raised over the same five-year period as the internal audit finding, as shown in Table 4.2.

Whilst there is not such a diverse range of topics as the internal audit data, there are similarities. Many problems have been found with the inventory list requirements, and the disaster recovery arrangements. The sub-contact problems related to acquired software, usually a bespoke project or a Commercial-Off-The-Self (COTS) product with a bespoke element. The problems found include companies not being selected from an approved suppliers list, no supplier assessment, and no management plan with any evidence of review. The software management plan has also been highlighted as problematic.

Table 4.2. List of topic and frequency of deficiencies found on 2nd party, customer software audits over five years.

Rank	Topic of Deficiency Found	Number of Non-conformances	%
1	Documentation errors	23	22%
2	Software inventory listing	21	20%
3	Disaster recovery plan	16	15%
4	Lack of review	12	11%
5	Sub-contract management	12	11%
6	Insufficient software audits	9	9%
7	Software management plan	8	8%
8	Test records	4	4%
	Total	105	

4.4. Software Audit Trail Investigation

This section sets out to investigate whether there is any correlation between the type of problems found and the audit trail likely to have been taken by auditors. Of interest is how thorough these audits have been, particularly in terms of the depth of the trail. In stepping through a typical audit, it would be reasonable to suggest that during the planning and the initial part of the audit, the person responsible for the

software inventory list would act as a point of contact (POC) and, indeed, would be audited early in the audit trail. The assessment of the appointment and the associated responsibilities contained within a job description would be the first requirements tested as part of the audit. This would be followed by an assessment of the existence and quality of the inventory list. Evidence that the list did contain all the software from that area and that all the listed data was correct and up to date would be checked. This would be followed by the assessment of the other main responsibilities - the approval, appropriateness and quality of the local management plan. From this point it would reasonable to suggest that the auditor would then assess the compliance of a sample of software projects or systems. It is also reasonable to suggest that the audit would follow a typical sequential software development life cycle for new projects. Using the waterfall life cycle the audit trail sequence is depicted in Table 4.3.

Table 4.3. Sequence of topics for a typical software system audit using the waterfall life cycle model.

1st	2nd	3rd	4th	5th	6th	7th
Respons-ibility	Inventory	Management Plan	Req. Spec.	Design	A coding standard	Testing

The audit may then assess other support topics such configuration management, disaster recovery, service plan and records of review. Table 4.4 shows the topics with most audit deficiencies from the internal audit findings results.

Table 4.4. The topics with most deficiencies from the internal software audit findings

1st	2nd	3rd	4th	5th	6th	7th
Inventory	Manage ment Plan	Disaster Recovery Plan	Document errors	Responsibility	Change control	Y2K

It can be seen the first three topics assessed in a generic software audit appear in the top five topics for the numbers of deficiencies found. If this is compared to a mature software system in a well-established maintenance regime, the audit trail would resemble that depicted in Table 4.5.

Table 4.5. Typical audit trail for a software system in the maintenance phase.

1st	2nd	3rd	4th	5th	6th	7th
Respons-ibility	Inventory	Management Plan	Change control	Service plan	Test Records	Disaster Recovery plan

Again we see a further correlation to strongly suggest that the topics with most deficiencies from the internal audit findings are the ones that would be assessed during the early part of a software audit. This suggests that auditors may be less inclined to continue with an audit once they have found a few deficiencies. The internal audit planning documentation was reviewed to establish any correlation of reduced scope for these software audits. This proved particularly interesting as there was not any indication of reduced scope for routine audits on the implementation of software quality system, but a number focussed audits on one software topic. These included disaster recovery, sub-contract management and Y2K, the millennium bug problem. Of further interest was that the audits on disaster recovery and sub-contract management pre-dated 2nd Party customer audits on the same topic by a couple of months. It appears that these were specially targeted internal software audits to hopefully reduce the numbers of problems found by the customer on their subsequent 2nd party audits. This explains the reason for the high numbers of deficiencies found on these specific topics and further corroborates that auditors are likely to raise one or two problems at the start of an audit, and less inclined to be thorough, once these have been found.

4.5 Auditor questionnaire

As part of a major research project on audit efficiency three questions were raised that relate to this investigation. They were designed to test the preparedness and motivation of auditors.

As these questions were to be sent via email they were initially piloted with one-to-one meetings to reduce any ambiguity and ensure clarity. After a slight revision they were sent as an email attachment with an explanation of the objective – that this was an opportunity to improve the audit process and the questionnaire provided them with a vehicle to have their say. Twenty-three auditors were sent the questionnaire, of which five were full time auditors, with eighteen conducting audits on a part-time basis. There is a requirement that these part-time auditors undertake at least one internal audit a year, as a minimum. The response rate was 100%. The overall average score for preparedness was high at 4.17. All auditors declared they were not under pressure to raise at least one non-conformance and only one auditor stated that they were less inclined to be as thorough having raised a deficiency. The questions were:

1. *Which statement reflects your preparation for an audit?*

Indicate	Statement of Preparation
X	
5	Detailed review of processes and procedures taking on-board output from planning, develop question set and strategy to test effectiveness.
4	Detailed review of processes and procedures, develop question set to test salient points and compliance.
3	Review of processes and procedures, identify mandatory requirements to select questions
2	Review relevant procedure and highlight all the shalls
1	Turn-up with the relevant procedures and ask questions based on mandatory requirements (shalls)

	Yes	No
2. *Do you feel under pressure to raise at least one non-conformance?*		
3. *Having raised a deficiency or two, are you less inclined to be as*		

thorough for the rest of the audit?

Unfortunately these results are pretty inconclusive, and a bit of a contradiction to the initial conclusions, in the audit trail investigation. Upon reviewing the results perhaps the questions were really asking how professional auditors are in their jobs. Human nature being what it is, the answers were then as you would expect. This prompted some face-to-face discussions and one group meeting with the full time auditors. These were not recorded to encourage freedom of expression. Under these conditions it did start to emerge that auditors would in fact feel they had met part of the audit objective by raising at least one or two deficiencies. Indeed, comments were made that there had in the past been criticism if at least one non-conformance is not found on audits. Also, some comments were made on the likelihood that some audits could be less thorough, having raised one or two deficiencies, but this would depend on the circumstances. Comments included; time constraints for an audit, the fact that part-time auditors may be less thorough than full time auditors, and also that auditing can really only sample work activities and will never be totally thorough. Although, this anecdotal evidence supports the conclusion from the audit trail investigation, perhaps this research proved to be more of a lesson in question setting.

4.6 Audit approaches - discussion

This investigation has raised questions on the various approaches and techniques used in internal audits. The possible inappropriate use of sampling techniques is highlighted, as is the need to use checklists to ensure a holistic approach is taken.

It has been suggested (Arter, 1989) that the decisions on sample technique should form part of the audit planning process. Care should be taken on deciding the sample technique itself, whether random from a range of systems or more objective in terms of importance. Indeed further investigation suggested it is a science in its own right.

Sampling can be probabilistic, or non-probabilistic, judgmental, blocked or statistical (DoE, 2003). To balance this, it is also suggested (Arter, 1989) that sampling should be based on just a percentage and indeed that too much time can be spent on the science of sampling.

The use of an audit checklist can be summarised (ISO/IAF, 2003) by its advantages and disadvantages.

Advantages:

- Aids planning
- Ensures consistent and systematic approach
- Is a memory aid
- Supports collection of evidence data and record
- Can save time if presented in advance.

Disadvantages:

- Can be intimidating
- May limit focus
- Is restrictive
- May hinder communication when relied on
- If generic, may become irrelevant
- Limits unique questioning and constricts the audit path.

The main use of a checklist (Taormina, 1999) seems to be the mandatory practice or requirement from a procedure or standard translated into a question. The use of standardised or generic checklists available from many commercial companies is not considered good practice (Arter, 1989). Apart from the non-applicability aspects, they may miss an important control element, crucial to the success of a system. Also, audit checklists have a role to play in measuring the effectiveness of a quality management system (Davidson, 2004) if they can be utilised as a self-assessment tool.

4.7 Conclusion

This chapter has provided an example of an internal auditing system not necessarily delivering on its intended benefit when assessing practitioners' compliance with a company defined software quality system. It may have been successful in complying with the ISO 9001 quality standard clause 8.2.2, for undertaking audits for software, but it has been ineffective in terms of realising real improvements. This has been highlighted by the analysis of the audit findings, and the subsequent investigation. In this respect there appears to have been a lack of a customer for the software audit, someone to conduct this analysis on a regular basis. This lack of customer focus may have influenced the behaviour of the auditors, in terms of lack of accountability and the thoroughness and depth of the audits undertaken. The use of many different part-time auditors may have impacted on the thoroughness of audits. It has highlighted that care must be taken with asserting that audits are only a sample. Taken in isolation, the fact that a non-conformance against the system has been found, can be meaningless. Some further analysis and understanding is required. The problem identified needs to be put into context, is this a minor slip-up or is this the opening of the proverbial can of worms? What is the extent of problems, and why are they presenting themselves? An investigation to provide an holistic approach to problem analysis is initially presented in Chapter 6 and further developed in Chapter 10.

Although this research focussed on a software quality system as its subject matter, the lessons learned are generic and could be applicable to other subjects for internal auditing. This research clearly identifies the need for an audit customer. This is someone who cares about, and is prepared to analyse the results. For software this could be a software quality manager. This is feasible for organisations that are software intensive, but otherwise these responsibilities could be rolled into that of a more general quality manager role. What would be important is the establishment of a system to test and then track the true level of implementation.

Chapter 5. An Improved Process Model for Internal Auditing

Chapter Preface

This chapter extends the research from Chapter 4 to further investigate and understand the reasons why internal auditing is often perceived to not add value. The need to develop a new process model and approach is proposed that will improve the actual and perceived value of auditing. An analysis of the process identified areas of potential inefficiency and conflict. A questionnaire was sent to auditors and auditees at AWE plc, to gain their views on audit effectiveness and quantify their perception of value.

This chapter is based on "An Improved Process Model for Internal Auditing" (Elliott et al, 2007d) published in the Managerial Auditing Journal, June 2007.

5.1 Introduction

The conducting of audits, internal to a company, to assess the status of a quality system is a regular occurrence, yet never seems to be popular nor necessarily results in a positive or beneficial outcome. It is often considered to lack respect (Small, 1998); indeed, recent events in the financial sector (WorldCom, Enron etc.) have done little to improve the reputation of the auditing profession. The negativity created by auditing can impact morale and adversely affect working practices. So it is a useful exercise to examine the nuances within the process, identify the elements that are ineffective or create difficulty, and the reasons why. The ineffective elements could then be eliminated, bypassed or their impact controlled, or new process elements developed. However, it is important to quantify the current status so that changes made can be measured to validate improvement.

It is traditional to think of an audit in terms of the annual checking of a company or business accounts. The financial compliance audit evolved in the 20th Century as business practices became more complex (Arter, 2000). Often the undertaking of a financial audit will provide an increase in confidence that the company's business is being conducted in an honest and appropriate manner. This enhancement of credibility was picked-up by the quality profession in the 1940s (Arter, 2000) and first applied to the military and then, in the 1960s, the nuclear power industry. The implementation of a quality inspection regime by the military was a reaction to accidental explosions of munitions in factories (Seddon, 2000). Indeed the recent history of auditing has often been heavily influenced by adverse events (Cangemi and Singleton, 2003).

Research for this chapter challenges some of the fundamental, possibly protected, influences on the internal auditing process (ISO 19011, 2002). Whereas it is important in external auditing to remain impartial and not supply any form of consultancy, this is not a driver with internal auditing. Auditors gain vast amounts of knowledge and experience from exposure to good and bad quality management systems. Yet this advice and consultation opportunity can be actually suppressed within auditing processes (ISO Guide 62, 1996), although it is acknowledged that within the financial sector, consultancy has been incorporated into internal auditing definitions (IIA, 2004). Research for this chapter focuses on management style internal audits, conducted by its own employees and used by companies as part of the demonstration of compliance to standards such as ISO 9001:2000. The questionnaire referenced in the methodology section was presented to staff at AWE plc.

5.2 Methodology

A review of the activities within the internal audit process was undertaken by mapping out the process, the top level is represented in Figure 5.1. This was developed, initially from first hand knowledge and experience, with reference to the

AWE internal audit procedures, standards and guidance, and the many authoritative books on auditing (Wealleans, 2000; Russell, 2001; Arter, 1989). This was conducted to a level where the potential inefficiencies or areas for potential conflict could occur. These issues were then tested in a questionnaire to validate whether there were indeed problems. Standards and guidance helped to identify the drivers and influences to internal auditing. Journal and conference papers, magazine articles and web sites were reviewed to ascertain the current trends, issues and concerns that currently surround internal auditing.

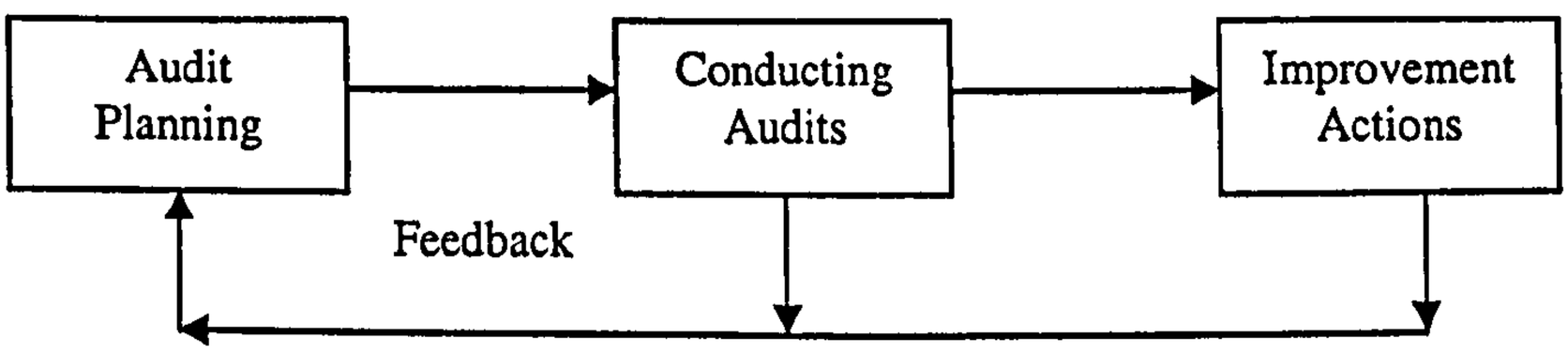

Figure 5.1. Top level process map of an effective internal audit process

Two questionnaires were devised, one for auditors and one for auditees. The questionnaires provided the recipients with an opportunity to comment on their overall perceptions on the value of the auditing process and state whether particular process elements were strong or weak. As the questionnaires were to be sent out via email, initial versions were drafted and a small number of face-to-face interviews took place, with both auditors and auditees, to test the instrument. After the initial alterations, a further pilot study was conducted via email. The email contained a message to explain the purpose of the questionnaire with suitable encouraging words to offer recipients the chance to voice concerns and improve the process. With minor modifications, the final questionnaire was sent out to all five full-time company auditors and a further 20 from the company's part-time auditor register. Twenty-five questionnaires were also sent to people known to have been audited in the previous two years. Then a random selection of a further 25 company employees were contacted. The return rate was an excellent, 100% from the auditors and 54% from

the auditees. The questionnaires contain some common questions so that the results could be compared and analysed from the two different standpoints.

5.3 Analysis of the internal audit process

Internal company audit schemes are often derived from the ISO 9001:2000 Standard clause 8.2.2, which stipulates the need (or mandatory requirement) for an internal audit process. The essence of the clause is that audits must be planned and organised. The plan should indicate that each clause in the ISO standard, as well as the elements of the company management system are to be audited. The type of activities considered important by the company should have a priority over the more mundane activities. The scope or depth of an audit may be influenced by the risks to the company. Information from previous audits should be considered along with what is termed 'planned arrangements'. These planned arrangements can relate to a programme of activities or actions stipulated in a management or quality plan. The output of audit planning would be a schedule or programme. These typically are managed over a yearly cycle. From the execution of the plan, a view could be formed that the generic auditing process has three distinct stages: planning, conducting, and reporting (Karapetrovic and Willborn, 2000). As there is a need to consider previous audit information in the planning stage, this could be considered as a feedback loop. This is consistent with the process described in ISO 19011, the principle guidance for quality system auditing. Here the emphasis is on the management of an audit programme.

Within a framework of Deming's: Plan-Do-Check-Act, methodology (Deming, 1986), the process stages are; authority for the audit programme, establishing (plan), implementing (do), monitoring (check), with feedback for improving the audit process (act). These process views could be considered self-serving in terms of effectiveness, as success can be perceived as the undertaking of the process and completion of the programme. This could account for some of the criticism found on auditing in that it

often fails to add value. The review of recent literature for this research found an obsession with ways to conduct value-adding auditing. Indeed, Greg Hutchins (Hutchins, 2002) of Quality Plus Engineering, has his web site URL named as "www.valueaddedauditing.com". The New York Stock exchange's Securities and Exchange Commission requires value-added audits on 17,000 listed companies. The Institute of Internal Auditors (IIA) has revised its definition of internal auditing to include "designed to add value."

When considering the effectiveness of an audit from an auditee's perspective with a system that truly adds value to the company, the findings and improvement actions take prominence. So from an effectiveness perspective the main process stages would be planning, conducting and improvement actions, as depicted in Figure 5.1. This depiction seeks to emphasise from the outset the importance of the improvement actions. The ISO 19011 standard does not provide an answer on what is an effective audit and how to measure its value.

5.4 Questionnaire Development

Part of this research seeks to document, what is an effective audit and how it can be quantified. The questionnaire was designed to capture the internal auditing perceived value at AWE plc. This was achieved for the overall process and some of the detailed stages. These serve as a baseline from which to track any improvements, put in place as a result of the recommendations from this research. Terms that describe an effective audit include "reliability of findings", "added value", and "client satisfaction" (Beckmerhagen et al, 2004), but these are outcomes and value still requires quantification.

Beckmerhagen et al (2004), state an effective audit:

- has adequately defined audit objectives, approved by all stakeholders

- has a suitable plan accepted by all
- allows for adequate resources; people and time
- is executed by competent auditors
- is conducted in accordance with a standard or procedure
- has findings that are valid and significant to record which are analysed against objectives and risk and that lead to improvements
- has satisfied clients
- provides evidence of improved working practices

The questionnaires developed are given in the next section, Section 5.5, which also gives the results of the questions. In the absence of a tangible and quantifiable value for internal audit effectiveness, question 1 provides a perceived value from both the auditor and auditee perspective. This is probably the most significant value from which to track audit improvement initiatives. This is similar to the IIA's survey question on measuring general respect (Baker, 2002). It is important to an auditee that they are given sufficient notice of an impending audit. If insufficient notice is given the audit becomes an inconvenience with negativity that is hard to erase. Companies with successful audit departments, such as DuPont, consider marketing and communication an integral part of the auditing process (Roth, 2000). Indeed, auditee buy-in should be initiated in the planning stage. Effective audit planning should result in sufficient notice to all auditees. Question 2 provides an assessment of good planning and communication. A successful audit is one in which preparation is crucial. Time should not be wasted on finding documents or evidence requested. Suitable preparation also indicates respect for the system and a positive approach to the process. Focussed preparation can also be from a result of the clarity of objectives and precision for audit investigation.

Evaluating how much time auditees spend on preparation, as conveyed in question 3, gives an indicator of these attributes. In a similar fashion, question 4 obtains a demonstration of the auditor's respect for the auditee's time that will be spent on the

investigation. An auditor with knowledge and experience is more likely to appear prepared and is more likely to identify problems. Question 5 captures an appraisal of an auditor's essential communication and interpersonal skills (Smith, 2005). This would include putting the auditee at ease to lessen the degree of stress (Vinten 1994) and in dealing with any conflicts to ensure a reasonably positive experience. Question 6, can also highlight successful interpersonal skills but more specifically it indicates whether any of the problems that have been raised are worth recording. There is historical data that demonstrates that auditors occasionally raise deficiencies that auditees consider trivial (Wealleans, 2000). It provides an indication that an auditee is comfortable that the issue has been found and that it will lead to improved working practice.

An important activity for making auditing effectiveness is to understand the factors that have lead to a non-conformance. If the underlying root causes are not dealt with, the problem is likely to re-occur. Question 7 records the amount of activity for this crucial part of the process. Similarly, question 8 measures a fundamental reason for undertaking an audit. If improvement actions are not undertaken well, the value is significantly diminished. Question 9, the feelings of auditors and auditees towards future audits, relates to, and validates the perceived value of auditing outlined in question 1. If audits have been of value and have made a positive contribution to working practices, motivation for future audits will be high.

Questions 10 to 15 were sent to just the auditees. From an auditee's point of view, a successful audit would be one in which they were aware of when it was to take place and its objectives, so they could prepare accordingly. The investigation would also be conducted in a professional manner, the auditor would be knowledgeable on the subject matter and they would discover situations that when corrected could improve working practices. Question 10, on awareness of the AWE Company audit programme, helps gauge the success of the planning process in terms of communication. It indicates the extent of the consultation and approval process. As

meeting audit objectives is one of the fundamental criteria for an effective audit, question 11 requests that all-important measure. Auditor competence is an emotive subject and difficult to quantify, but a good way to ensure that appropriate auditors were being appointed would be for the client to rate how knowledgeable the auditors appeared during the audit. This is the objective of question 12. JC Penney (Roth, 2000) puts auditors in the stores during the holiday seasons to get a firsthand look at the business needs, ensuring they know the business and how it works. As mentioned by Beckmerhagen et al (2004), an effective audit is one in which findings are valid and significant to record and action. Question 13 seeks to rate the significance as perceived by auditees. It will help reduce the risk of trivial issues being raised as outlined in question 6.

Once the audit findings have been reported or communicated, the resulting actions are the responsibility of the management area that has been audited. An insight into the effectiveness of this sub-process aligns to the overall value of the auditing process itself. If a local management area is spending time and resources monitoring resultant actions, the perceived value of the findings would be of some significance. Often company systems (as in AWE) require central tracking of audit deficiencies. Unfortunately, this can impact adversely on the quality of closeout where the objective becomes the closeout not the improvement. Questions 14 and 15 give an insight into these areas.

5.5 Questionnaire: Results and Analysis

<u>Section 1. Common questions to both auditors and auditees</u>

Q1. Overall Effectiveness

1. Overall how would you rate the effectiveness of the audit process?

No value (1)	Little Value (2)	Some Value(3)	Moderately Effective (4)	Effective (5)	Very Effective (6)

Auditor Result (1)			Auditee Result (1)		
Average Score	Variation	Answer Range	Average Score	Variation	Answer Range
4.00	Low	3-5	3.45	High	1-5

This auditor score of 4 out of 6, and the auditee score of 3.45 out 6, are important results for the current perception that both auditors and auditees have for the overall value of the internal audit process. These values could be tracked in future questionnaires to demonstrate that changes made to the auditing process from this research, are indeed improvements. It is a fair result which probably reflects the acceptance that audits are necessary in business and that they certainly should add value. Fairly predictably, the auditors' perception of their work was greater then the auditees. The auditee result could be considered more valuable, as these are the main customers for the process. In this respect, it is their views that should be used as a benchmark as it is a more accurate reflection in the standard of audits being undertaken. The variance value does indicate a high inconsistency in the perception of the quality of auditing.

Q2. Notice to Auditees

2a. How much notice (typically) do auditees' actually receive?

Phoned and it was a surprise (1)	Day before (2)	Days before (3)	Week before (4)	Weeks/Month before (5)	Know the programme(6)

2.b When you were audited how much notice (typically) did you receive?

None, just turned up (1)	Day before (2)	Days before (3)	Week before (4)	Weeks/month before (5)	Know the programme(6)

Auditor Result (2a)			Auditee Result (2b)		
Average Score	Variation	Range	Average Score	Variation	Range
4.61	Low	3-6	4.50	Low	3-6

The notification system is a tried and tested method of communicating the impending audit. Sufficient time should be given to organise meetings and arrange interviews. But the notification should act as a reminder of the earlier audit planning. The scores

4.61 out of 6, and 4.50 out of 6, averaged to the result of "a week before" and would indicate that this system is acceptable.

Q3. Auditee Preparation

3.a In general how much effort do auditees make to prepare prior to an audit?

None (1)	A little (2)	Some (3)	Reasonable (4)	A lot (5)

3.b How much effort do you make to prepare prior to an audit?

None (1)	A little (2)	Some (3)	Reasonable (4)	A lot (5)

Auditor Result (3a)			Auditee Result (3b)		
Average Score	Variation	Range	Average Score	Variation	Range
2.83	Medium	2-5	3.40	Medium	1-4

The results show an element of disagreement between what the auditor believes has occurred and what the amount of preparation the auditee has actually carried-out. However, both results present a less than satisfactory indication, as an auditee should be reasonably prepared for an audit. This may be an indication that audit objectives are not as clear as they could be.

Q4. Auditor Preparation

4.a Which statement reflects your preparation for an audit?

Indicate	Statement of Preparation
X	5. Detailed review of processes and procedures taking on-board output from planning, develop question set and strategy to test effectiveness.
	4. Detailed review of processes and procedures, develop question set to test salient points and compliance.
	3. Review of processes and procedures, identify mandatory requirements to select questions
	2. Review relevant procedure and highlight all the shalls
	1. Turn-up with the relevant procedures and ask questions based on mandatory requirements

4.b In general, how well prepared do you feel auditors are?

Not at all (1)	Little (2)	Some (3)	Prepared (4)	Well prepared (5)

Auditor Result (4a)			Auditee Result (4b)		
Average Score	Variation	Range	Average Score	Variation	Range
4.17	Low	3-5	3.45	High	2-5

The auditor score is rather predictable as it represents a statement on their professionalism. Auditors seem to feel auditees do not prepare for the audit as well as they might and this is substantiated by the less than reasonable rating. The variance suggests auditees have witnessed occasions when an auditor appears unprepared.

Q5. Interview Atmosphere

5.a In general how would you describe the atmosphere during an audit interview?

Very hostile (1)	Hostile (2)	Guarded (3)	Reasonable (4)	Open & positive (5)

5.b How would you assess the manner in which the audits are conducted?

Very Poor (1)	Poor (2)	Reasonable (3)	Good (4)	Professional (5)	Very Professional(6)

Auditor Result (5a)			Auditee Result (5b)		
Average Score	Variation	Range	Average Score	Variation	Range
3.83	Low	3-5	4.00	Medium	2-5

The results of 3.83 out of 5, and 4 out of 6, are reasonable and consistent scores. Additional comments that were provided by recipients in the questionnaire indicated that auditors do try to make the exchange a positive one, and lessen the degree of stress created by the audit.

Q6. Problem Acceptance

6.a How do auditees feel about accepting deficiencies found?

Very annoyed (1)	Annoyed (2)	Slightly annoyed (3)	Accept (4)	Fully accept (5)

6.b How do you feel about accepting deficiencies found?

Very annoyed (1)	Annoyed (2)	Slightly annoyed (3)	Accept (4)	Fully accept (5)

Auditor Result (6a)			Auditee Result (6b)		
Average Score	Variation	Range	Average Score	Variation	Range
3.48	High	1-5	3.45	High	1-5

Again the auditors and auditees agree, although variations in response appear significant. This could be due to a number of reasons, certainly the more trivial issues raises such as documentation errors will create resentment, as will any disagreements between the auditor and auditee. Resentment can increase if the reporting mechanisms focus on the number of non-conformances found and not the impact that these issues have on business effectiveness.

Q7. Audit Deficiency Analysis

7.a How much analysis of the reasons for the deficiencies do you feel is undertaken?

None (1)	A little (2)	Some (3)	A lot (4)

7.b How much analysis (root cause) of the reasons for the deficiency is undertaken in your area?

None (1)	A little (2)	Some (3)	Reasonable (4)	A lot (5)

Auditor Result (7a)			Auditee Result (7b)		
Average Score	Variation	Range	Average Score	Variation	Range
2.30	Medium	1-3	3.05	High	1-5

This result highlights an area where the overall value of the internal audit process can be significantly diminished. Without proper root cause analysis the underlying reasons for the problem may not be dealt with. In turn this could result in the problem re-occurring which severely reduces the impact that a good audit should have. The high variance again highlights the inconsistencies in undertaking this activity.

Q8. Quality of Improvement Actions

8.a How would you rate the quality of closeout for deficiencies?

Very Poor (1)	Poor (2)	Reasonable(3)	Good (4)	Very good (5)	Excellent (6)

Auditor Result (8)			Auditee Result (8)		
Average Score	Variation	Range	Average Score	Variation	Range
2.83	High	1-5	2.90	High	1-4

This is a truly significant result that fundamentally questions the overall value of auditing. The low scores on both sides provide clear indication that the immediate improvement actions are not being carried-out as they should, let alone root causes being attended to. Correcting this issue needs to be the focus of improving the internal audit process.

Q9. Motivation for Future Audits

9.a How do you feel about conducting an audit again?

Very Negative (1)	Negative (2)	Okay (3)	Positive (4)	Very Positive (5)

9.b How do you feel about being audited again?

Very Negative (1)	Negative (2)	Okay (3)	Positive (4)	Very Positive (5)

Auditor Result (4a)			Auditee Result (4b)		
Average Score	Variation	Range	Average Score	Variation	Range
3.78	High	2-5	2.95	High	1-5

The auditor result of "okay" is perhaps rather predicable. More significant is the more negative response from auditees. This is likely to be related to the overall perception of value.

Section 2. Specific Auditee Questions

Q10. Programme Awareness

10. How aware are you or have you been of the Company audit programme?

Totally unaware (1)	Aware but never seen it (2)	Aware but never reference it (3)	Aware and occasionally reference it (4)	Aware of dates I'll be audited (5)

Average Score	Answer Variation	Answer Range
4.00	Low	3-5

The result was good considering the regular suggestion to increase the visibility of the system. Launching the programme with presentations and news articles would increase visibility.

Q11. Audit Objectives

11. When you have been audited, how clear were the criteria for a successful audit?

Not at all (1)	Poor (2)	Some understanding (3)	Good (4)	Totally clear (5)

Average Score	Answer Variation	Answer Range
3.05	High	1-5

This is perhaps a disappointing result and it certainly identifies the need to convey expectations of success or otherwise of an audit. This may also be a reflection that the reporting system focuses on the non-conformities and does not provide an indication of overall performance, which may be very good.

Q12. Auditor Knowledge

12. In general, how would you rate the auditors' understanding of the subject matter in your area of interest?

None (1)	Poor (2)	Some (3)	Good (4)	Very good (5)	Excellent (6)

Average Score	Answer Variation	Answer Range
3.30	High	2-5

The high variation to the answer to auditor knowledge may be indicative of the high part-time auditor turnover. This would be something that needs to be managed by the audit manager when selecting auditors during planning.

Q13. Significance of Findings

13. In general, how would you rate the significance of findings?

Totally insignificant (1)	Very minor (2)	Minor (3)	Of some significance (4)	Significant (5)	Very significant (6)

Average Score	Answer Variation	Answer Range
3.60	High	1-5

This result can, in-fact, be interpreted in two ways. Local areas may be following best practice and only minor issues exist, or auditors are only finding minor issues. A number of "totally insignificant" scores were recorded, which would be of concern.

It would not be possible to correlate the answers to question 12, auditor knowledge to that of the answers to question13 significance of findings as the specific auditors were not named nor their understanding investigated. It could also be considered intrusive.

Q14. Improvement Monitoring

14. How much monitoring occurs to track improvement actions?

None (1)	A little (2)	Some (3)	Reasonable (4)	A lot (5)

Average Score	Answer Variation	Answer Range
2.90	High	1-4

The low score may again indicate that the local management area audited felt that the impact of the original deficiency was not important.

Q15. Close-out Pressure

15. How much pressure are you under to closeout the deficiency?

None (1)	A little (2)	Some (3)	A lot (4)

Average Score	Answer Variation	Answer Range
2.95	High	1-4

If there is relatively little monitoring of improvement, it is unlikely that local management will be applying pressure to closeout. It is likely that any real pressure would come from a central organisation wishing to clear its database of corrective actions.

5.6 Discussion

It is evident from the responses to the questionnaire that the internal auditing process is very well known and generally perceived as being necessary. Yet behind this acceptance there are many concerns people have in the way audits are conducted. The questionnaire results cast doubt on the overall value and added benefit of auditing and in general this has led to the negative attitude towards it. Of particular concern were the answers to questions 8 and 9, the analysis of problems and the quality of improvement action to closeout deficiencies. These are two key issues that greatly impact the effectiveness and value of auditing. They are actions that are under the direct control of the management of the area that has been audited and demonstrates a weakness in the general management of this important interface in the auditing process.

The main driver for the internal audit process comes from the need to meet the ISO 9001:2000 Standard clause, 8.2.2, that supports the attainment of gaining, as Vinten (Vinten, 1998) describes, "comfort certification at an acceptable price". Although it is implicit that opportunities to improve are realised through the identification and action to address problems found, it is clear that this has not been totally effective, and at times quite trivial. The system has also been found to be open to abuse, particularly at the verification stage. If the ISO clause 8.2.2 were to state the audit

process had to demonstrate its own effectiveness then this would significantly change the focus. The phrase "Doing it for the ISO" (Wealleans, 2000) is often heard as the objective of undertaking an internal audit, however to be truly effective, audits require far more beneficial objectives. Audit investigations are excellent opportunities to provide advice and guidance on working practices. In third party certification situations this is seen as providing consultancy (ISO Guide 62), and may adversely affect impartiality. It is interesting to note that the additional explanation from the International Accreditation Forum (IAF, 2003) aggressively discourages any form of consultancy. From the evidence that consultancy-type activities are not described or discussed in authoritative, internal auditing books or standards (eg. Arter, 1989 and ISO 19011), it would be reasonable to state that this advice and guidance aspect has also been eliminated from internal auditing schemes.

5.7 Recommendations and improvements

The most significant change that is recommended as a result of this research is that the principal focus of conducting internal audits is changed. The current aim of internal auditing should be transformed from ensuring that the programme or schedule is met, to be one of capturing and tracking the actual benefits and savings resulting from it. This changes the associated assumption that the problems found will lead, hopefully, to improvements being made to a new focus that would state that internal auditing has eliminated poor practice and initiated savings of some quoted estimated value.

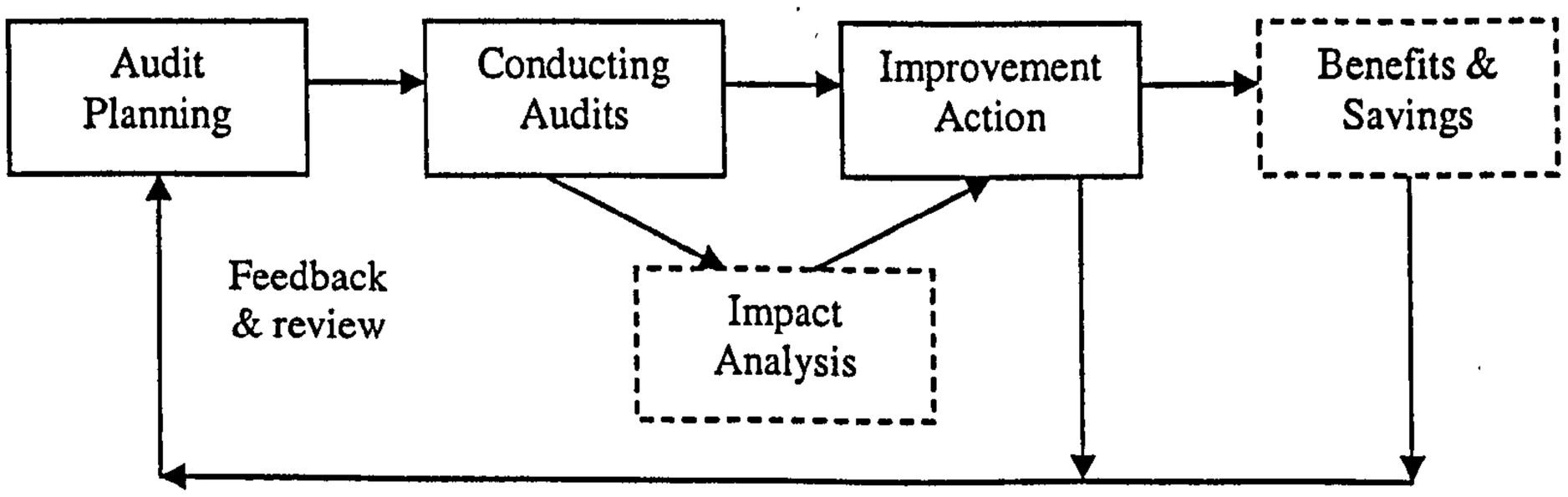

Figure 5.2. Improved top level process map for internal auditing

This could be achieved, with the development or collation of cost models or industry best practice statements for the effectiveness or otherwise of various processes. These would be referenced or applied during the conducting phase to determine what is, or what has been the impact of a problem found on business practice. Descriptions of the various scenarios and consequences would be stated, and then cost estimates of poor practice recorded. It will be unlikely that there will be the data available to provide a totally accurate figure, but approximate calculations with margins of error would prove useful. The important activity is to investigate the problem fully with all the contributing factors and assign associated costs. This will facilitate the recording of subsequent benefits and future cost savings as a result of correcting of the problem and the underlying causes. This may also add impetus to the analysis of wider company issues and trends associated with the original problem. This, in turn, may identify further company savings.

It would be reasonable to expect the above improvements to eventually have a significant impact on the numbers and types of non-conformities. To monitor the effectiveness of the new audit process, the tracking of the number of problems found per audit or unit of audit activity should be established. There should also be a reduction to the number of occasions the same type of problem is found. This is in contrast to the usual measure associated with internal audits, the number of audits conducted against the plan, which merely indicates that the audit process is being carried out. It is likely that this approach will eliminate the finding and recording of trivial issues that have blighted the auditing process (Wealleans, 2000). The visualisation of the internal audit process would be modified from that of Figure 5.1, to that in Figure 5.2. These changes should ensure that the audit process is seen in a more positive light by those audited. If every audit is remembered for the benefits

and savings made, the auditees will start to think of an audit as a helpful measure rather as an intrusive process that hinders their work.

5.8 Further research

Models or checklists could be developed to capture local management behaviour and working practices, to obtain a true understanding of why compliance has not been achieved. The reference cost models will be used to assist in estimating the cost impact and other implications of non-compliance. This will facilitate the estimation of the cost-benefits of implementing the agreed improvement actions. Collation of this data will answer the question of how much the company is saving as a result of conducting audits. A pilot study of this approach is presented in Chapter 10.

5.9 Conclusion

This research concludes that internal audits are not always well received, they are sometimes not performed well, and the findings are not always particularly significant, and at times trivial. Root cause and trend analysis are not sufficiently utilised to help eliminate future deficiencies. The improved process model and Cost-Benefit Audit Methodology (C-BAM) approach was found to be unique within the scope of the literature review. The implementation of the recommendations from this research will significantly change both the perceived and actual value of internal auditing, as its cost-benefit will be recorded.

To fully understand the extent of the problems uncovered in this chapter a detailed analysis is required of the state of software quality system implementation within the company, and this is covered in the next chapter.

Chapter 6. An Analysis of Software Quality Management at AWE plc.

Chapter Preface

A detailed question set is required to test and measure the true extent that a software quality management system is adopted and implemented across a large company like AWE plc. The analysis of the gathered data reveals specific topics of weakness that can also reflect the cultural acceptance or resistance that management groups have towards the adoption of quality systems. Having identified detailed problems and barriers, effective strategies and programmes can be deployed to improve the level of implementation and, therefore, the effectiveness of a software quality management system. This chapter presents the question set used and the subsequent results obtained from the implementation assessment for 55 software systems at AWE plc. These data are collated into management groups and the associated cultures discussed. The topics of weakness are highlighted together with the very specific actions that are least undertaken. A range of improvement actions is also presented.

This chapter is based on "An Analysis of Software Quality Management at AWE plc" (Elliott et al, 2006b) in the proceedings of the Software Quality Management (SQM) conference in April 2006, and in Volume 15 of the Software Quality Journal (Elliott et al, 2007a).

6.1 Introduction

The two principal aims of this research chapter are, firstly, to ascertain how well adopted and implemented the software quality management system is at AWE plc, and secondly, to gain an understanding of any problems or issues that were acting as a barrier to the implementation of the system. The case study utilised data collated during 1999, from the implementation assessments of software systems across the

company from many different managerial areas. These areas and their associated software projects were responding to the requirement to meet the software quality management systems as defined in the Company Software Procedures.

This research has been facilitated by the appointment of the author to the position of Software Quality Manager at AWE plc and the motivating factor was to baseline current practice so that any improvement achieved from subsequent initiatives could be tracked and credit claimed. The intent was to put in place improvement initiatives after gaining a good understanding of any problems and the reason for them. The responsibilities of the position of Software Quality Manager include: maintaining the defined software quality management system; reviewing and approving the management plans of important, high risk, software products; maintaining a list (register) of known software items across the company; and ensuring implementation is regularly checked through audit or assessment. These data presented in this chapter have been collated from undertaking these audits and assessments.

The software quality management system at AWE plc had been an integral part of the company's management system since 1991, and ISO 9001 certification was first successfully achieved in 1994. The software quality management system had received a high degree of attention, highlighted by its explicit reference in the top-level scope of the certification. This high level of profile is consistent with the importance that software has in the research and development of nuclear weapons. Implementation of the software system would be considered mature as frequent assessment by the customer, the Ministry of Defence (MoD), has occurred every six months, over a subsequent five-year period. Earlier research (Elliott., et al 2005) had discovered some flaws in the way implementation had previously been checked, particularly in the internal audit process. The use of audit deficiencies did not provide a useful tool for understanding the issues and problems associated with implementation.

6.2 Methodology

To meet the objectives of this research, a measurement system was devised to check and assess the adoption and level of implementation of the software quality management system, and also provide a means to gather information on specific topics so that problem areas could be identified. The assessments were then conducted, primarily as part of the 1999 internal audit programme, but also in an ad-hoc fashion when some guidance or clarification on the software quality management system was requested. As the assessments were conducted, any comments that related to general feelings towards the software quality management system, the role of local area software representative (LASR), and any attitudes towards being assessed, were recorded. This led to an impromptu survey on the LASRs as to their understanding and motivation about their position and duties. Three basic questions were asked of the 70 LASRs via email; (1) Did they know what to do? (2) Did they know how to do it? (3) Did they want to do the job? They were then asked if they would like some or further training for the role. The survey results are captured in Table 6.7. The assessment data were then collated by topic area and by main functional group to see if there were specific improvement opportunities or cultural issues that needed addressing.

6.3 The Measurement System

The measurement system devised to capture the true level of implementation of the software quality management system was known as an implementation rating. The system was a checklist of specific questions covering the entire software quality management arrangements and was divided into topics. Each topic has a series of questions relating to implementation tasks and graded 0-4, as shown in Table 6.1. The first four levels, 0-3, are a hierarchy of compliance, and the last, level 4, known as "established" relates to the maintenance of the systems and in particular, the

documentation. Most of these data were gathered as part of the Company audit programme.

Table 6.1 The defined levels of the implementation rating.

Level Heading	Criteria and Description
0. No evidence	Unaware of requirement, no documentary evidence of implementation available
1. Some Evidence	Aware of requirement, some historical evidence available, less than half the implementation requirements for that topic have been met.
2. Reasonable	Evidence of historical implementation, may not be up to date, more than half but not all the implementation requirements for that topic have been met.
3. Compliant	System fully implemented, all requirements met, but may be immature
4. Established	Has been reviewed and subsequently updated, in place for something like a year and is now institutionalised

It is a well known observation that prior to an audit, the area to be audited will suddenly make an effort to put in place many of the requirements they are to be audited on. Many auditors find the appropriate documents requested have often just been published with "the ink still wet." Clearly these are immature systems and not normal practice. It is therefore quite likely that these newly created documents will not be maintained. So the "level established" is a check for evidence that the defined local arrangements have been in place for a while, have been used and updated for changes, are adding value and provide a true reflection of actual practice.

The subject areas in which the company software quality management system was divided are listed in Table 6.2. These subject areas were chosen on the basis of importance. Although a topic area such as the categorisation system did not contain many implementation requirements, the decision on this category does, as it assigns the subsequent required control activities.

Table 6.2 Subject areas of the software quality management system assessed by the implementation rating.

Software Quality Topic	Explanation
Organisational responsibility (org)	LASR appointed with responsibilities in job description
Software Register (reg)	All software systems recorded with baseline data
Categorisation (cat)	Level of importance assessed, management life cycle defined
Management Plan (scp)	Management and control arrangements defined in a plan
Specification (spec)	User and software requirements specifications
Design, development & coding (des)	Design information and applied coding standard
Reviews & verification (r&v)	Life cycle stage outputs reviewed and verified
Testing & validation (t&v)	Test plan, specification procedures, records arrangements
Configuration Management (scm)	Identification of items & baselines, change control, status accounting, audits
Disaster Recovery (d.r.)	Data & system back-ups and recovery plan
Computer Services (c.s.)	Service plan, measures, configuration management
Procurement (proc)	Specification, contract controls and testing arrangements

6.4 Results and Analysis

Tables A II 1. to A II 8 in Annex A II are the detailed results of the software quality management system assessments conducted in 1999. Each table represents the data collated by each of the main organisational functional groups. For each system assessed the level achieved against the implementation rating (Table 6.1) was assigned as that value for each software quality topic (Table 6.2). The totals for that topic and the average value are presented for each topic. The overall total value and the maximum achievable are also presented so that the functional area percentage implementation can be recorded and these are collated in Table 6.3 together with the overall Company average. Table 6.4 presents the overall Company average for each

software quality topic to indicate the strengths and weaknesses of implementation areas.

Table 6.3. Average assessment results for each of the functional groups

Functional Group	Average % Implementation Rating
Production Groups:	
Weapon (weap)	49.46%
Infrastructure (inf)	43.62%
Technology (tech)	38.72%
Research (pers)	33.15%
Support Groups:	
Safety, Quality, Security (sqs)	28.7%
Commercial (com)	27.5%
Finance (fin)	26.74%
Personnel (pers)	25%
Overall Company Average	**34.1%**

The assessment sample of 55 systems is sufficient to provide a high degree of confidence in the interpretation of the results. It was rare to record a zero rating for a software quality topic for each system assessed, which demonstrates that nearly all areas visited had made some attempt to comply. The low occurrence of rating values of three, to indicate full compliance, signifies a lack of drive by local management to achieve full implementation of the software quality management system, and there was certainly an absence of any systematic check. Of further concern is that the benefits of implementation are not being sustained as the compliance evidence of software quality activities is being poorly maintained, as indicated by the even fewer number of level four ratings. Documents or outputs from activities are at times not being referred to, used or updated when changes occur. Initial implementation to achieve compliance is neither being established nor institutionalised.

As can be seen from Table 6.3, an overall company performance implementation result of 34.1% has been achieved for the software quality management system. The result could be considered low, which raises questions as to why the certification to ISO 9001 had not identified the problem, and why had it never been raised by the subsequent continued assessments or the internal audit process. This low score can, in-part, be explained by the method of gathering the information. The audit process requires evidence, mostly documentary, that the various processes required actually took place. If this is not available, or is insufficient, the assessed score will be low. So good practice may well have taken place, however, not the best practice as the required proof was not retained as evidence. This provides an example of Weinberg's (1993) relativity of quality. Quality to the auditor is shown by good documentation, whereas quality to a software developer is represented by good functional programs.

Table 6.4 Average assessment results for each topic across the Company

Software Quality Topic	Average Company Implementation Rating
Organisational responsibility (org)	2.03
Categorisation (cat)	1.98
Management Plan (scp)	1.69
Testing & validation (t&v)	1.6
Software Register (reg)	1.56
Disaster Recovery (d.r.)	1.43
Procurement (proc)	1.33
Specification (spec)	1.06
Computer Services (c.s.)	1
Configuration Management (scm)	0.78
Reviews &Verification (r&v)	0.72
Design, development & Implementation (des)	0.6

A list of the strongest and weakest topics, as assessed and averaged out across the whole Company, can be seen in Table 6.4. The weakest implemented topics, together with the list of least undertaken activities in Table 6.5, provide a succinct training needs analysis, highlighting the topics requiring significant education and supporting guidance documentation.

Commenting on the more extreme values, on the plus side the appointment of staff to support implementation has given 'organisational responsibility' the highest rating but this hides the motivation issues uncovered in the LASR survey in Table 6.7. The categorisation scheme is well known and a cause of discussion highlighted by the comparative good score. The lowest scoring topic of 'design and development' revolves around a number of issues. Firstly, software life cycle management is not well known. Decisions on what type of life cycle to use and when are not understood and neither is the need to have design information for medium and long term software maintenance. It is apparent that most software developers are designing "in the code". The quality assurance topic of 'review' was poorly implemented as the understanding of types of reviews and the evidence of any formal activity was poor. There was, however, anecdotal evidence that informal reviews had taken place. Of serious concern was the low scoring on specification. Many aspect of specifying requirements and requirements management were missing on this critical practice.

Table 6.5 Activity least undertaken for each topic heading.

Software Quality Topic	Main Problem
Organisational responsibility (org)	No evidence of a local review for the need of local procedures, or check of implementation.
Software Register (reg)	Local registers do not contain all required entries
Categorisation (cat)	Under categorisation of software items and no evidence of categorisation review
Management Plan (scp)	Not all required topics covered in the plan, and the quality of the information was generally lower than expected
Specification (spec)	Incomplete and out of date specifications
Design, development & Implementation (des)	No review of system design characteristics and the appropriate life cycle to follow
Reviews &Verification (r&v)	Very little evidence of review activity
Testing & validation (t&v)	Test plans are generally not maintained, and poor traceability of test records to software previous versions
Configuration Management (scm)	Little planning and development of baselines, status of CIs not captured or maintained, no CM audits
Disaster Recovery (d.r.)	Insufficient testing of the recovery plan
Computer Services (c.s.)	No monitoring and poor maintenance of service plans
Procurement (proc)	Little evidence of the review of requirements

Table 6.6 List and frequency of comments made during assessments

Comment	Support Groups (com, sqs, fin, pers)	Research Group (res)	Other main Groups (tech, inf, weap)
It's not clear what the requirements are	Often	Often	Occasionally
We need to read and understand all of the procedures before we now what to do	Occasionally	Often	Occasionally
Don't understand terminology	Often	Often	Rarely
I can't visualise what needs to be done, can't you provide a diagram?	Often	Occasionally	Rarely
It's not what we do	Rarely	Often	Rarely
It's bureaucratic	Often	Often	Occasionally
That's not our responsibility	Often	Occasionally	Rarely
We're just not doing it	Occasionally	Occasionally	Never
Who are you to tell me how to do my job?	Never	Occasionally	Never

The group comments recorded in Table 6.6 have identified a significant relationship between culture and resistance to implementation: the more frequent and critical the comments, the lower the level of implementation. This is corroborated by the finding that the supporting areas (sqs, com, fin, pers), are more critical of the system and have lower scores than the main production groups (weap, inf, tech, res). Culture is created by personalities that have been influenced by their interests, training, knowledge and experiences (Spencer et al, 1992). There is an indication of an unwillingness to comply in these support groups, as shown by their comments. A key theme is their expectation or hope that other groups, the Information Technology department or sub contractor, will do the quality assurance (QA) for them. Any verbal agreements were not documented and sometimes not actually agreed.

The science and research group (res) has been singled out in the comments list of Table 6.6, because of the slightly different type of comments provided. Of particular

interest is the occasional recording of "Who are you to tell me how to do my job?" This conveys a degree of animosity, which will adversely influence their willingness to comply and does explain the groups placing at the bottom of the main functional group list.

Many of the software systems being assessed in this research group were providing and supporting leading edge science and technology. The software developers were themselves, highly qualified research scientists. However, they would not, in general, have had any formal training in software engineering, and the terminology of the software quality management system was not well understood. Indeed the term "engineering" applied to their work was not welcomed. They would, in general, be the users and the developers of the software. In this respect, the quality goal of customer or user satisfaction is dealt with, as it is an obvious personal desire. The requirements are also well known, as the functions to be programmed are science based and are less likely to be misinterpreted. Motivation to be productive is also likely to be high, as it supports an interest and desire to advance science. The risks presented are not realised until a scientist moves on or leaves the company. Then the detailed requirements and the design of the software are lost, because they have not been documented. An implemented software quality system would have ensured the design knowledge of the individual was captured, so in cases where it is not, the potential benefits of the quality system are then not realised.

In contrast, the weapons group was far more willing to accept Quality Assurance (QA) as an integral part of their job. In general, there was a good understanding of what was required and there were no problems with terminology. This group had a high number of qualified software engineers. Similarly, the infrastructure area (inf) demonstrated a high degree of willingness to achieve compliance but not always a full understanding of what was required. The technology group (tech) has a mixture of both scientific research and engineering staff so, from the analysis of the impact of

cultural influence made so far, its position in the implementation hierarchy between the infrastructure group and the science research group is not too surprising.

Table 6.7. LASR survey results out of 70 people questioned

Knew the totality of the responsibilities	Knew how to perform in the role	Wanted the appointment	Wanted training
19	9	23	33

Table 6.7 provides a clear indication that the role of LASRs was a poorly understood in terms of responsibilities and how those responsibilities need to be carried-out. The detailed survey results indicated that no one met all three main questions. i.e. if they knew what to do and how to perform the role they didn't actually want it. Some responders, although they did not fully understand the role, did not wish to be trained for fear of the position becoming more permanent.

6.5 Analysis of Detailed Results in Annex II

The assurance results from Table A II.1 show that the organisational arrangements attracted the highest scores on the implementation rating. Within this topic, the actions of appointing of LASRs with appropriate job descriptions were generally undertaken well. All staff had set-up a local register but their lack of understanding of configuration management techniques meant that these listings were of limited value. With only one level three and no level fours recorded across all systems, a complete lack of systematic checking of compliance was evident.

As the commercial group only owned two software systems, as documented in Table A II.2, it would be difficult draw definitive conclusions. As procurement is a main function of this group, it is not too surprising the one system scored well in this area

for software. Knowledge on the need to back-up data in case of a disaster was evident by the higher scores on this subject.

Table A II.3 shows that assignment of the correct category for software systems attained higher scores. It became apparent during the assessments that a number of meetings had been arranged to agree the category assignments and this was an area where two level fours were attributed. Unfortunately, the conducting of reviews for other life cycle activities did not score highly. On other topics, scores were better, such as the production of control plans.

The financial group scored highest on testing and validation as can be seen from Table AII.4 . The people in this group have an affinity for getting the numbers right and this is reflected in by their highest score. A lack of technical knowledge resulted in low levels being recorded on most other subjects.

Predictably, the personnel group scores in Table A II.5 were highest for the organisational arrangements which include appointment of LASRs and sign-up to job descriptions. During the assessments it became evident that they had relied on the Information Technology department to provide most of their software quality documentation. Unfortunately this re-assignment of responsibility had not been documented in local service-level agreements which accounted for some of the low scores.

The eleven systems assessed in the technology area provided the biggest variance of results. Total absence of specification and review records, together with a lack of design, accounted for the main problems areas, as can be seen in Table A II.6. The appointment of LASRs had been completed and the production of control plans were reasonable. The categorisation system was also quite well understood. The poor use of registers and the configuration management process present a clear training need in this area.

The highest system assessments were recorded in the Weapons group captured in Table A II.7. Unfortunately the lack of knowledge and ownership of one software system recorded a complete zero result. This highlights the stringent audit process that requires documentary evidence "at the time of audit". The main concern for this group was the significant risk associated with the loss of data that should have been documented in disaster recovery plans. This had been completely overlooked on many systems. However, this was an organisational group where the maintenance of the quality system was good and seen to add value, as emphasised by the reasonable number of fours recorded.

The research group has some very large and complex software and the need for design information was a hot topic of debate, which Table A II.8 shows scored low. Risk factors associated with staff turn-over and the learning time needed to understand complex software were considered in the discussion on reliability and maintainability. The skill and ability of some scientists to understand complex structures was astounding as was the motivation to develop software to advance scientific knowledge. These attributes compensate to some degree for their lack of software engineering knowledge.

6.6 Discussion on ISO 9001 certification

The overall assessment result leads to the question of what implementation level the software quality management system is actually required to achieve ISO certification. This question is, in fact, unfair because a level is not obtained as part of the assessment. The assessment process consists of two main elements, a review of the documented management system to ensure it meets the ISO 9001 standard's clauses, followed by a site visit to assess implementation. During the checking of implementation, the auditors will review the documented practice to that actually

performed and, if there are discrepancies, issues will be raised. These issues are generally graded in terms of severity of deviation from the intended practice. These grades or criteria will be documented within the third-party's own certification procedures. An example would be:

- an observation - for a slight discrepancy,
- a minor non-conformity - when a standard clause is not fully met,
- a major non-conformance - when a standard clause requirement is not complied with, or a significant number of related minor non-conformities have been found.

The initial certification depends on not receiving a major non-conformity during the site visit. If one or more have been received, certification is deferred until they have been addressed. This requires a follow-up site assessment, three months later, to ensure the appropriate corrective action has taken place. So, although a significant amount of compliance is needed, the actual "pass" for certification could depend on the way the company deals with specific corrective action. A similar, but slightly more relaxed situation exists after the initial certification, with continued assessments. These normally occur at six monthly intervals. In these circumstances, when a major non-conformity is found, a three-month period of grace is allowed before certification is removed. Again this presents a crisis management situation with specific actions to be addressed. This point was raised with one-third party ISO 9001 auditor, with 17 years auditing experience. The answer to the specific question of how many companies was he aware of, that had failed a third party ISO 9001 assessment, was none. Only one company had failed at the documentation review stage of the ISO 9001 certification process.

Also of consideration is the depth of audit concept (Elliott et al, 2005). This suggests that there is a reduced likelihood of an issue being raised, or certainly a major issue being raised, if a number of system elements, which may be administrative in nature, are in-place at the start of the audit trail. In this respect a typical audit trail for the site assessment of the software quality management system consists of:

1. a review of the company software quality management system documentation.
2. an assessment of responsibilities of a company point of contact, the software quality manager, who can demonstrate:
   - where software is in the company, the register of registers,
   - that high risk management plans are independently approved,
   - that the software quality management system has been audited.
3. the assignment of a point of contact for each management area with:
   - an inventory, the register of software in that area,
   - a control plan to demonstrate how the software is managed in that area.

This presents a scenario of a significant amount of audit checking, for administrative type activities, before the "quality" of implementation is assessed in a local area.

6.7 Conclusions

There are a number of highly significant conclusions that can be made from this research, indicating a successful achievement of the objectives. The overall implementation level of the software quality management system was felt to be too low, and this has led to the provision of support for improvement initiatives. The collation of results by main functional groups provides clarity of where support is most needed. The results facilitated an investigation into why the internal audit process (Elliott et al, 2005) had not previously identified the low level of implementation. It also questioned the role of ISO certification, in view of the fact that AWE plc has a valid reputation for producing world class software and is at the forefront of science and technology. This validates Weinberg's assertion (1993) that any software culture can be considered successful. Of concern, but of less profile, is that the compliance element of what is implemented is not considered to be valuable enough to be maintained properly, as demonstrated by the sparseness of the 'institutionalised' implementation rating of four. This indicates a lack of drive and attention to detail to achieve it.

The results do show a commitment to achieve implementation and that there are some well established elements of the software quality management system: the appointment of a LASR, the register, the categorisation system and the main management plan. However, it is the actual quality of the implemented system that needs to be improved.

Comments received during the assessment clearly indicate that cultural resistance has had a significant impact on adoption and subsequent implementation. Culture is often deep seated, and this will probably be quite a difficult problem to overcome. Some of this resentment emerged as a consequence of the original initiative to meet the first ISO 9001 certification in 1994. Indeed there was, at times, a resentment of the audit assessments as well, typified by the frequent comment, "Who are you to tell me how to do my job".

The reasons for the low level of performance can be summarised as a lack of understanding and the resulting cultural barriers. Training needs can be ascertained from the lowest scoring topics in 6.3 and the least undertaken tasks in Table 6.4. These are: requirements management, development and use of design, how to conduct reviews and software configuration management. Learning will be sustained with the provision of guidance documentation on these topics. The maintenance of documentation is a major issue and this is due, in part, to the lack of understanding of configuration management. When changes occur, many baseline documents are not up-dated, because they do not form part of a baseline listing, so the impact analysis of the change is incomplete. The appointment of people to the role of LASR has not been totally effective as indicated by the responses to the questions in Table 6.7. If someone is not happy with being appointed to such a position, they are unlikely to perform it well.

There is correlation between these issues found at AWE plc to those highlighted in the reports and standards in the literature review. The weapons group (weap) and the infrastructure group (inf) fit Weinberg's (1993) "steering" culture view that they are dealing with their quality problems, whereas the science research group (res) exhibit some of the behaviours of the "oblivious" culture in that they are sometimes unaware they are performing software development, but also the super programmer element of the "routine" culture. Finally the support groups can be at the "oblivious" and "variable" levels, if they know they have quality problems, but they do not know why.

Training to improve understanding is the key improvement action due to the prolific developer and user role, particularly in the science research area. Conveying the principles of software engineering and how they relate and how they can improve current working practices, should be the focus of this training. There is also a need to provide some training for senior management to convince them of the benefits of a software quality management system and to facilitate the drive and the much-needed systematic checks.

6.8 Recommendations and further research

An important conclusion that can be drawn from this research is that the system needs to be reviewed and revised to provide a clearer message as to what is required. In this revision, extensive consultation is needed to address the ownership issue. Any revision should retain the well-established system elements, such as registers, LASRs, a categorisation system and a local management plan. There are similarities between identifying configuration baselines and an inventory listing so these should be integrated in change control procedures, i.e. the register should capture the entire baseline and then be used to identify the configuration items that are impacted by that change. Training in configuration management and conducting reviews, particularly

design reviews, are key areas for improvement. A competency framework for the various roles for software management, such as Manager, User or Developer, should be devised with suitable training programmes. The support groups need to become knowledgeable customers and research scientists need help to understand the principles of software engineering and how the system can provide benefits. A training scheme which included an evaluation of learning effectiveness is outlined in Chapter 8. It would be useful to develop a self-assessment version of the checklist, to possibly facilitate local management review and audit, which again will help with the ownership and the systematic checking issues. This approach is described in Chapter 7. It would also be of interest to carry out research to learn from other companies failing ISO Certification and the reasons why, but it may prove difficult to obtain the data and hence is beyond the scope of the research presented in this thesis.

The implementation rating measurement system has demonstrated that it is an effective gap analysis tool and it has identified the specific technical issues that need to be addressed to improve implementation. The results and comments gathered during assessments also identify a relationship between the level of achievement and cultural resistance to the software quality management system. This chapter has shown that the cultural aspects need to be understood in developing an effective strategy for improvement. The various approaches to improve implementation from the baseline results presented in this chapter are investigation and reported in Chapter 9.

Chapter 7. The Effectiveness of Self-assessment.

Chapter Preface

Self-assessment can be an invaluable tool in which to measure, enable and monitor improvement to any management system. The flexibility of a self-assessment approach is one of its key strengths. Self-assessment can be utilised as an integral part of a major company Total Quality Management (TQM) programme, or initiative to achieve a recognised quality award, such as systems associated with the European Foundation for Quality Management (EFQM). It can also be effective in a more localised, low-profile, "bottom-up" approach. Outlined in this chapter are some pitfalls in its use and how they can be minimised. It is asserted that full adoption of self-assessment is not achievable as adoption barriers exist: these are discussed. Part of the aim of this chapter is to review the effectiveness of the use of a self-assessment methodology, as used, to measure the implementation level and facilitate the improvement of the Software Quality Management System (SQMS) at AWE plc.

7.1 Introduction

The actions required to implement an effective software management system, as stipulated by the Company Software Procedures, were captured and documented in a Microsoft Excel Spreadsheet. The purpose of this was for it to be utilised as a self-checking tool for local business areas to assess their own level of implementation. The entries appeared as a series of questions, firstly to state applicability, then to verify that the requirement had been implemented.

The system was developed in part, to deal with a number of problems associated with incorrect behaviours resulting from the internal audit process (Elliott et al, 2005) highlighted in Chapter 4. It had also been identified that local business areas were not

systematically checking that the Company requirements were being addressed (Elliott et al, 2006b), discussed in Chapter 6.

The self-assessment approach was encouraged by the author at regularly held presentations on the software quality management system. At an establishment like AWE plc with six thousand permanent staff and many contract staff, there is a need to maintain awareness of the company quality systems, sufficient to keep pace with staff turnover. The benefits of self-assessment conveyed included:

- It is a quick, easy and systematic way to check SQMS implementation
- It provides an automatically calculated percentage for implementation
- The results can be collated into topics to identify strengths or weakness
- It produces easily identifiable actions for improvement
- It facilitates root cause analysis to deal with all contributory causes
- It provides a baseline result from which to monitor improvements and foster the philosophy of continuous improvement.
- Self-assessment reduces the need for scarce Company audit resource (Waina,2001).

These marketing points above closely resemble the self-assessment definition provided by the EFQM (1999) which states:

"Self-assessment is a comprehensive, systematic, and regular review of an organisation's activities and results referenced against the EFQM excellence model. The self-assessment process allows the organisation to discern clearly its strengths and areas in which improvements can be made and culminates in planned improvement actions, which are then monitored for progress."

The investigation element of this chapter documents and analyses why self-assessment is used and adopted in some cases and not in others. Motivation for this research is, in part, due to an observed recent decline in its use, particularly for training purposes, as can be seen in Table 7.3.

7.2 Methodology

A Microsoft Excel spreadsheet containing the requirements of the software quality system were captured as questions-sets for self-assessment for a number of topics as can be seen in Appendix II. The question-set was a further development of the audit check list described in Chapter 4. Also, the levels, 0-4, of the implementation rating system presented in Chapter 5 were condensed to two percentage levels based on compliance to 75%, for levels 0-3 and up to 100% at level 4, which continued to be described as the "established level" thus retaining the initial structure of the measurement system. The software quality management topics included; general management, requirements and configuration management, design and coding, testing and validation, support to the operational use of the software system and other support topics such as disaster recovery, computer services and procurement. The use of the self-assessment tool was promoted on the company web site and in most communication exchanges. When a practitioner sought guidance on any aspect of the software quality management system, they would be encouraged to undertake a self-assessment so that a full examination of implementation could be conducted. Further guidance or support, would also be offered. It was also suggested, but not mandated to be undertaken, as part of the registration process for the company software training course.

The self-assessment spreadsheet was sent out to anyone who made an enquiry or sought advice about the software quality management system. Upon return it could be analysed for problems and level of understanding, as well as capturing the initial percentage implementation result. Further, supporting documentation could also be requested and reviewed with comments and guidance provided. When used as part of the training registration, the information on the attendee's local practices, strength and weaknesses in software engineering and quality topics could be gleaned, then elements of the subsequent training course could also be tailored to the specific training needs identified in these exchanges.

There are, therefore, various levels to the use and support of self-assessment at AWE. These range from areas using the system from their own initiative, areas that receive some guidance on terminology and subsequent actions, to areas receiving extensive support including document review, detailed guidance, training and monitoring of actions to achieving improved working practises and a subsequent increase in percentage implementation. The self-assessment spreadsheet could be used directly from the web site, without any feedback required by the user. However, when a practitioner sought advice and guidance then an improvement project was initiated. It was also utilised as part of the training registration process when it was encouraged prior to participation so that the course could be tailored to some specific needs.

During the time self-assessment was being encouraged, a number of exchanges would conclude without a spreadsheet being returned or any further exchange or support. Although the number of non-returns was not formally recorded, it was in the region of 25%. The comments, excuses or reasons given for not completing the self-assessment were captured and responded to. Typically, there would be an immediate comment relating to time, yet people would often promise they would "get around to it". Further chasing would reveal additional concerns. It was often the case that some attempt had been made to investigate what the self-assessment would entail, but there had been some further barrier to complete the task. Further comments could include being put off by the terminology, not understanding the question, and concern how the data will be used. Other more revealing comments were that the self-assessment would not be relevant or it may initiate too many unnecessary actions.

It was noticed that over a five-year period, the number of people conducting the self-assessment, as part of the training course and in general, was dwindling, see Table 7.3. So during one of the training courses, in which not many delegates had undertaken self-assessment (where they could be considered a captive audience) a questionnaire was given out to all seventeen delegates to investigate the barriers to completion as described in Table 7.1.

Table 7.1. Barriers to conducting self-assessment questionnaire

Your Name (optional)	

Please rank the applicable issues that were a barrier to you completing the self-assessment	Rank 1st, 2nd etc
Didn't feel it was totally relevant	
Couldn't really be bothered	
No recognition for doing it	
Worried about the implications – use of data	
Didn't really have time	
Just wasn't a priority	
Didn't really understand it	
It was too long	
It wasn't going to change what I do anyway	
Others not listed	

Would you be more inclined to complete the self-assessment if:	Yes/No	Rank
You had more time		
There were better instructions		
You received a £25 voucher to do it		
You received a £25 voucher to do it and achieved a 75% score		
The data was totally confidential		
If you knew before hand it was really going to make a difference		
If it was a mandatory company requirement		
Your boss told you to do it		
You knew you were was going to get support for improvements		
Your boss told you to do it and he didn't get the results		
Someone helped me e.g. 1-2-1 meeting, presentation		
Others not listed		

	More / Less	
Having attended this course, do you feel you are more or less likely to complete the self-assessment?		

These questions were promoted as not an exhaustive list, as just an initial suggested sample of ideas. Delegates were encouraged to consider and list other issues. The

delegates were asked to rank each issue in order of their main barrier to completing the self-assessment. The priority rankings assigned by the delegates were summated as a total score for each listed barrier, so a first priority would count as a one, and so on. The results, which can be seen in Table 7.2 indicate that time was the main barrier. All delegates stated they were more likely to conduct the self-assessment now they had attended the course. A further six weeks lapsed without a return of any subsequent self-assessment. Follow up phone calls revealed that "not being a priority" was the true barrier, followed by it not being either part of what they did or indeed that it has not been requested by line management.

Table 7.2 Barriers to conducting self-assessment questionnaire results

Rank	Most significant barrier	Score		Rank	Motivating factor to improve adoption	Score
1	Time	27		1	More time	28
2	Priority	35		2	Better instructions	46
3	Relevant	41		3	Help 1-2-1	51
4	Understand	43		4	Make difference	57
5	Too long	53		5	Boss	68
6	Implications	66		6	Support	70
7	No change	69		7	Mandatory	73
8	Recognition	73		8	Confidential	79
9	Not bothered	76		9	Boss, no results	83
				10	£25	87
				11	£25 @ 75% completion	89

Table 7.3. Take-up rates of people conducting self-assessment as part a training course.

Year	Number registering for training	Number completing and returning the questionnaire	% Adoption
2001	23	16	70%
2002	Data not gathered	Data not gathered	
2003	15	10	67%
2004	30	17	56%
2005	33	16	48%

7.3 Conclusion

Whilst the self-assessment methodology and the way it has been used have proved to be reasonably successful, it is evident that other strategies are needed to increase its adoption. The barriers to self-assessment have been captured and analysed to give a better understanding of the problem and enable subsequent adoption strategies to be devised. It is important to state that although other approaches will increase the number of people undertaking self-assessment, full adoption is unlikely without a significant driver for change. Formal audits would be one such example, in which the self-assessment was mandated and this is tested in Chapter 10.

Key to increased adoption of self-assessment is the involvement of line or senior management. The two major issues, reasons or excuses, of insufficient time and priory, will be dealt with if the local business area line manager can be persuaded to become involved or indeed have the assessment findings formally presented to them. The involvement of an independent person would help reduce response bias, however further commitment would be guaranteed if all the senior management were involved. Clearly explaining how the data will be used is important to allay fears, stressing a no blame culture. The requirement for clear instructions on how to use the self-assessment tool has also been highlighted. On a similar theme, simplifying the language or making it more compatible with regularly used terminology would also be useful.

In conclusion, the self-assessment methodology provides a further adoption strategy to the audit methodology investigated in Chapters 4 and 5. How self-assessment supported the training scheme is in conveyed in Chapter 8 and how this approach was used to monitor the overall improvements from the range of adoptions strategies is presented in Chapter 9.

Chapter 8. Providing Demonstrable Return-on-Investment for Organisational Learning and Training

Chapter Preface

There is growing agreement that one of the primary drivers, if not the key driver of long term organisational effectiveness, is the ability of an organisation to learn effectively (Sandi, 1996, Tennant, 2002, Hale, 2003). Learning organisations that anticipate, react to change and learn, are likely to maintain a competitive advantage. These organisations are constantly looking for more effective and efficient ways of training. Paradoxically, other organisations will often slash training budgets in times of hardship, as training departments are unable to demonstrate the effectiveness of their programmes.

This chapter is based on "An Providing Demonstrable Return-on-Investment for Organisational Learning and Training" (Elliott et al, 2007c) in the proceedings of the International Conference of Software Process Improvement Research into Education (InSPIRE) in April 2007.

A holistic approach to training is presented, that clearly demonstrates cost savings with improved effectiveness and efficiencies that are aligned to business objectives. Extending Kirkpatrick's evaluation framework (Kirkpatrick, 1998) with Philips's Return-On-Investment concepts (Phillips, 2002), this chapter conveys a number of successes, including trainee satisfaction and the capturing of improved knowledge and skills. It then goes on to give case studies of how, and to what extent this knowledge has been applied with examples of resulting efficiency savings.

8.1 Introduction

A training programme for teaching the concepts of software engineering and related support practices was introduced into AWE plc in response to a need identified from a

comprehensive software quality management system (SQMS) implementation assessment (Elliott et al, 2006b) and outlined in Chapter 6. The assessment uncovered a lower level of adoption of the SQMS than would normally be expected for an ISO 9001 certified company. This was due, in part, to a lack of knowledge and understanding of some of the software quality and engineering principles. A characteristic of this organisation is that a significant proportion of software development is performed by scientists and engineers in specialised units who, although they are highly qualified and experienced, have not been trained in software engineering but have undertaken software development to support their main role. In this respect, many software activities appear like a cottage industry and the analysis of the training need was relatively clear.

This chapter describes the training methodology adopted and how it fitted in to an overall strategy to improve the level of implementation of the SQMS. The research for this chapter has provided an opportunity to compare the methodology used against industry best practice evaluation techniques and to consider modern learning concepts to further improve the learning experience provided. A key theme within this research is the demonstration of how the training and subsequent learning is applied by delegates in their working practices and how the new knowledge and skills improve the efficiency of work activity and how this leads to a cost-benefit. The difficulties associated with demonstrating a true return-on-investment are conveyed with a few case studies on how this was achieved.

8.2 Methodology

The philosophy for the required training identified at AWE was to provide a good introduction to the principles of software engineering and link this into how to implement the company software procedures. A partnership was established with a training provider that not only had a ready made software engineering course, but also

had experience of implementing the methods and techniques from a diverse range of software systems similar to AWE. The author contributed to the training by providing the links to the Company Software Procedures. A feature of this part of the training was to not only provide examples of good AWE practice but also to identify some situations where people had misinterpreted the requirements in a manner that was slightly humorous. This helped give the course a lighter feel as software engineering can otherwise be a dry subject.

As described in Chapter 7, prior to attending the course, delegates were encouraged to undertake a self-assessment. This consisted of answering questions contained in a spreadsheet, on what actions required by the software quality management system had and had not been implemented. Self assessment was particularly useful in support of the training as not only did it baseline the current level of implementation from which to track subsequent improvement and value, but it also helped gain an insight as to the level of understanding of software engineering or quality processes. This helped tailor the training to specific individual training needs and present examples that were more relevant to attendees.

From the onset it was felt important to ensure attendees felt the course was of value. This was achieved in part through the monitoring of course evaluations but also by informal comments requested at the end of each course tutorial. The evaluations consisted of a comprehensive set of questions on course objectives, value, applicability, joining instructions, food, etc., to be answered on a six point scale ranging from 1 for excellent to 6 for very poor. This proved a good system and fully met the "reaction" element of Kirkpatrick's evaluation framework at level 1, described in Table 8.1.

After a number of courses it became clear that the AWE specific elements were considered the most valuable, so the course was altered to become a completely bespoke course, tailor-made to the company software system. Nearly all tutorials were geared to further implementing the SQMS requirements. Attendees would

document their own processes, collate inventories and assign risk categories and software product baselines. They also learned to understand system measures and metrics and apply their own. The courses were not made compulsory but the skills provided by the course were documented in a competency framework within the company software procedures.

Table 8.1 The achievements of the AWE plc SQMS training against the evaluation frameworks of Kirkpatrick (1998) and Phillips (2002).

Evaluation Level	**Training Objective**	**Achievement Action**
1	Reaction/Satisfaction	Very good course evaluation results and encouraging comments gathered after each tutorial or exercise
	Planned Action	Increased percentage implementation of the SQMS monitored from follow-up self-assessments. However, not achieved or considered by delegates not participating in self-assessment
2	Learning	Monitoring of correct actions from exercise and case study. Course test results.
3	Behaviour/Application	Increased percentage implementation monitored from follow-up self-assessments and discovered on internal and external audits.
4	Results/Business impact	Measuring the efficiency of applied learning. Gathering data on business benefits.
5	Return-on-Investment ROI	Calculation of cost-benefit from conducting training internally. Development of cost models to facilitate gathering of data on systems to demonstrate ROI.

The change to a bespoke internal course provided an immediate cost-benefit. The cost incurred from the partnership with the training provider, which amounted to £600 per delegate was no longer needed as the course was delivered entirely by the author. The course provided an introduction to software engineering principles which, to obtain this from an external company as single one-off course would cost in the region of £1500 per person for a four day course. The same amount and type of information and some exercises were put into three days of the AWE course, as there

was an element of "padding" in the external training. Part of the feasibility to change to a bespoke course included a review of course content from a range of training providers. The cost-benefit of providing an internal course is presented in Table 8.2.

Table 8.2 Training Course Cost-Benefit for an Internal Course

Issue Year	2000	2001	2002	2003	2004	2005
Number of attendees	14*	23*	n/a	30	30	33
Total cost savings	£12600	£20700	n/a	£45000	£45000	£49500

* £900 saving per delegate, due to cost incurred from having the training partnership for the years 2000 and 2001. The savings for 2003 onwards are calculated as £1500.

8.3 Measuring the Training Cost-Benefit

Follow-up assessments were conducted to monitor how much of the SQMS was being applied. This would appear as an increase in percentage in the self-assessment and would be in line with Kirkpatrick's (1998), model at level 3. As any improvement in the implementation of the SQMS can be directly attributed to the training provided, the benefits of the improved implementation of the SQMS can also be considered to be the benefits of the training when calculating the return-on-investment.

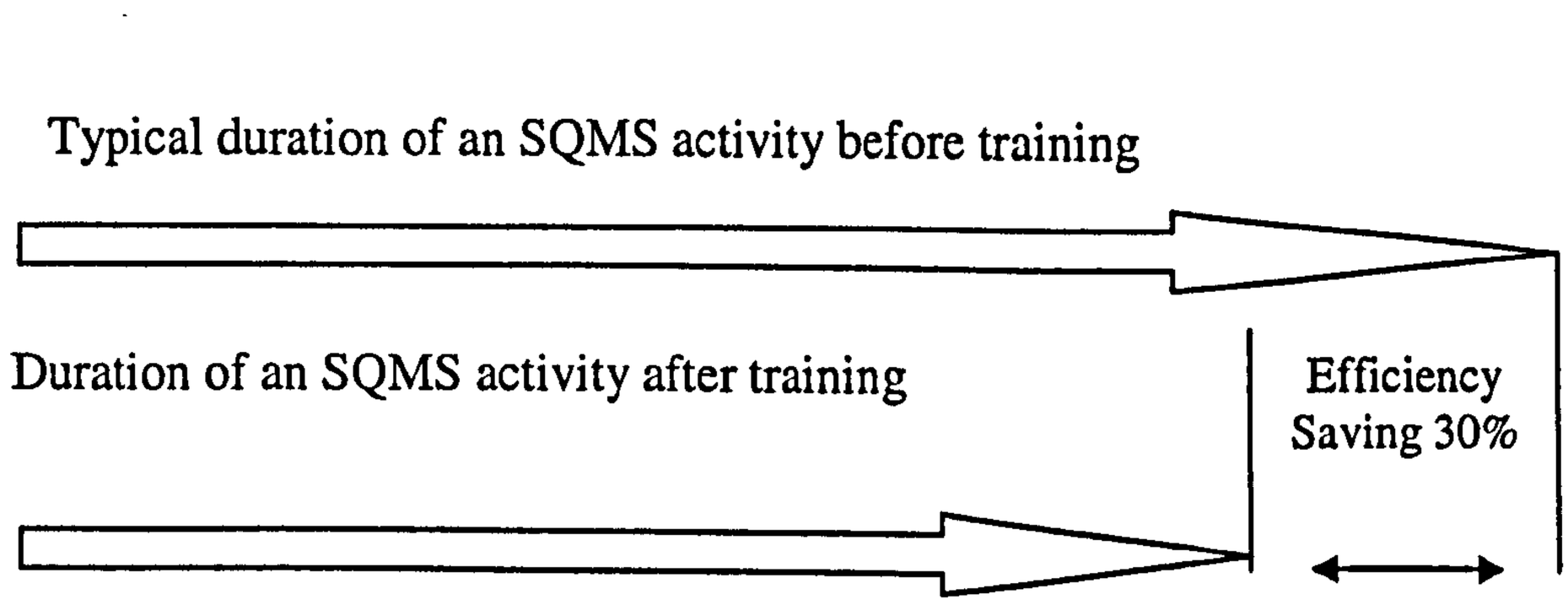

Figure 8.1 AWE plc training efficiency model

The first step to calculate the monetary value or cost-benefit associated with the training was to develop an efficiency model. The main beneficial impact of the training would be a significant reduction in the time some of the actions required by the SQMS would take to be completed. Taking the production of a software management and control plan as an example, delegates on the training case study syndicate would develop the main elements of a plan and have it peer reviewed in one and a half hours. The improved skills and knowledge acquired from this experience was found to significantly reduce the amount of time delegates took to produce a management plan in the future, giving a reduction of the order of 30%. This concept is modelled in Figure 8.1.

This increased efficiency could be modelled with some degree of accuracy as a reasonable amount of data were available from corrective action from audits to calculate an average value (Elliott et al, 2005). This concept was to develop a complete model for all the other learning activities that could take place from the training course to produce a total efficiency value. Further examples of efficiency savings include the reduced time for developing baselines, capturing requirements in the company template, assessing risks, deciding appropriate life cycle, etc. It is estimated that a total efficiency value of £15,000 could be realised if all SQMS activities were improved. For each reassessed system, the percentage improvement from before and after training was assigned against that percentage of the total efficiency figure to document the cost-benefit from the training as can be seen in Table 8.3.

Table 8.3 Training Course Efficiency Saving

Issue **Year**	2000	2001	2002	2003	2004	2005
Number of attendees	14	23	n/a	30	30	33
Attendees completing self-assessment	n/a	16	n/a	20	17	16
Adoption %	n/a	70%	n/a	67%	56%	48%
Average % improvement	n/a	19%	n/a	23%	15%	16%
Applied equivalent efficiency savings	n/a	£2850	n/a	£3450	£2250	£2400
Total efficiency saving	n/a	£45600	n/a	£69000	£38250	£38400

The next step for calculating the total return-on-investment (ROI) was to calculate the costs for the time on the training course for both the delegates and tutor, and the costs for other resources used in the training. Two further issues were brought into the cost-benefit and ROI equations. Firstly, the cost of applying or implementing the SQMS itself and the benefit the introduction that the new software system would have on, say, resource utilisation. A literature review on cost models was conducted to see if there were models available. The conclusion was there is really no generic model that can be used, although a number of cost statements exist on the various software engineering processes. These can be divided into two main areas, costing associated with new projects or costs associated with maintaining existing systems. It was decided to develop generic models that had some credibility for these two areas to facilitate discussions on cost-benefit. The first model was based on the Standish Factors for successful software or information technology project (Standish 2001). The factors were translated in the various software project management and software engineering processes. These were calculated using the cost of project overrun for an average IT project of 178% (Standish, 2001).

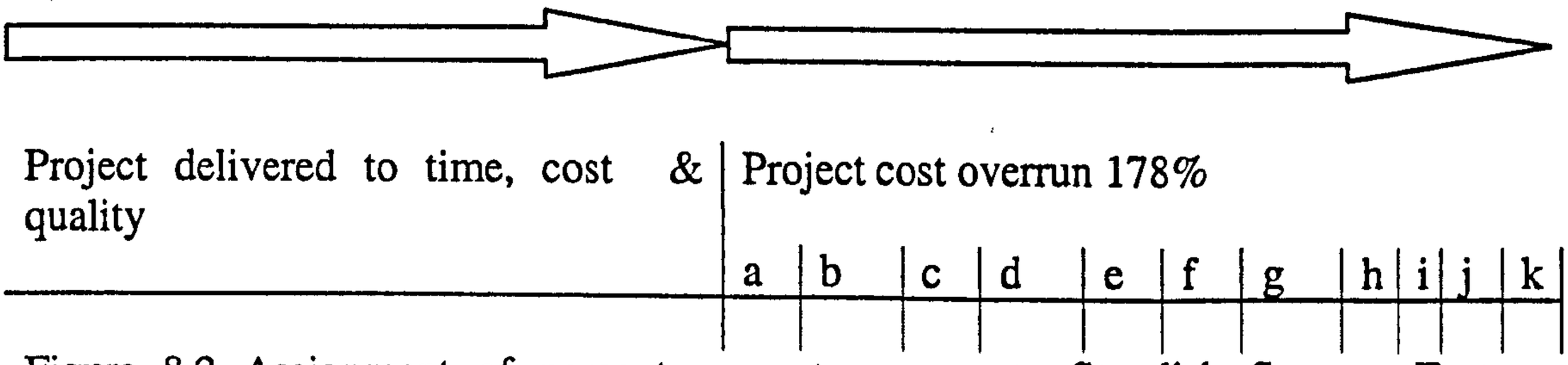

Figure 8.2 Assignment of percentage cost overrun to Standish Success Factors (Standish, 2001). (see Table 8.4 for key)

The concept for the model is that if none of the processes are implemented the total overrun of 178% would be applied. If all processes were implemented the project would be brought in on time to the allocated budget without any overrun. This is presented by Figure 8.2. A similar concept was applied for maintenance activities, in that each change is considered a mini project and the same project cost overrun of 178% could be applied. If all best practice software maintenance processes were

utilised no overrun would be incurred. If none of the practices were applied, the 178% overrun cost would occur, as depicted by Figure 8.3. These concepts were applied as a multiplication factor for each fulltime equivalent on the project. The idea was that these costs would help engage practitioners in discussions on efficiency and cost-benefit for their working practices.

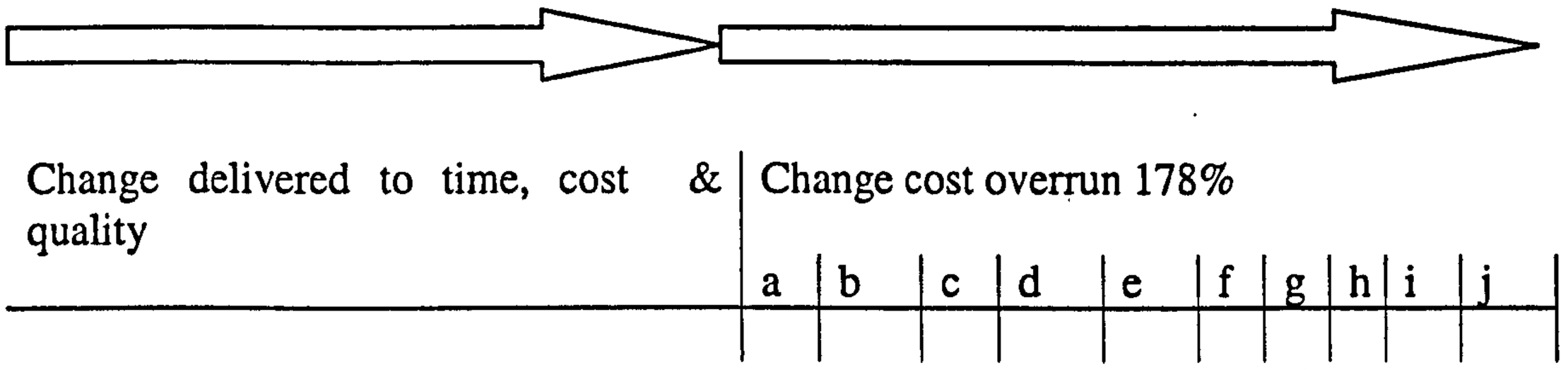

Figure 8.3 Assignment of percentage cost overrun to typical change activity

(see Table 8.4 for key)

Table 8.4 Key to Figures 8.2 and 8.3, Standish Project Delivery Success Factors and best practice maintenance processes.

Initial Project Delivery			**Maintenance (a change project)**		
Item	Standish Success Factors	%	Item	Typical SQMS Process	%
a	User Involvement	15.9%	a	General Management	5%
b	Executive Support	13.9%	b	Staff Effectiveness	15%
c	Clear Statement of requirements	13.0%	c	Requirements Management	5%
d	Proper Planning	9.6%	d	Design Techniques	20%
e	Realistic Expectations	8.2%	e	Programming Best Practice	10%
f	Smaller project milestones	7.7%	f	Testing Techniques	10%
g	Competent Staff	7.2%	g	Conducting Reviews	5%
h	Ownership	5.3%	h	Process Measurement	5%
i	Clear vision and objectives	2.9%	i	Configuration Management	15%
j	Hard-working, focused staff	2.4%	j	Effective Use & Efficiency	10%
k	All Other Practices	13.9%			

A validation exercise was conducted to see how accurate the efficiency model was against actual examples. Interviews took place which highlighted the problem that people were either not aware how long activities took or they were reticence to convey that information. A typical scenario would be that a person would say that to produce a Software Management and Control Plan would take a week in total duration, yet the actual duration could be many months. Clearly other tasks would be attended to, however in some cases this would be the highest priority task. Further probing and detailed investigations found that the average or typical values in the model were reasonably accurate.

Project benefits generally consist of the quantitative value, payback or interest that is returned for an investment. The costs for a project refer to the expenses, expenditures, and capital outlay necessary to apply the project, which will hopefully result in some benefit. The equations used to calculate the return-on-investment start with a benefit-to-cost ratio.

Benefit-to-cost ratio = Project Benefits/ Project Cost

The return-on-investment percentage is similar to the cost-benefit ratio except that the costs of the project are subtracted from the benefits before dividing by the costs.

Return-on-Investment = (Project Benefits - Project Cost) / Project cost

Another way the implementation of a SQMS can provide a benefit is through staff turn-over. Documents are often written and only used at that time of production or review. However, additional value can be realised in the event of staff recruitment, as the length of time needed to learn, understand and become proficient with a system would be reduced if comprehensive documentation was available. Costs could be compared in the following situations:

- How long would it take to learn what the software did without a requirements specification?
- How long would it take to understand the software structure without design information?
- How long would it take to replicate testing arrangements without a test plan, procedure, and specification and revalidate test data?
- How long would it take to understand what the software system consisted of without a baseline listing?

The case studies in the next section give two examples of how costs and benefits were calculated and the value demonstrated for the training courses and the software quality management procedures taught.

8.4 Return-on-Investment Case Studies

The first case study to be considered for a return-on-investment was a product management system. There were two aspects that were considered for calculating value and return-on-investment. Firstly from the efficiency of having the software system developed and then the benefit of producing the required documentation in the situation of staff-turnover when a new recruit had to learn what the system consisted of, and to become proficient in its management and use.

This system started life as a complex Excel spreadsheet that contained the entries of the components of a system. It was used to calculate production and maintenance schedule for in-service items. The spreadsheet was used to manually calculate a range of scenarios for optimum component production and/or maintenance. These scenarios could take many weeks to complete. It was decided to automate this process with a software system as part of a development contract. The developers of this Excel spreadsheet system had not applied any formal SQMS requirements and as

a consequence a cost of £45,000 was incurred to produce a requirements specification in a total project cost of £215k. To manage a new contract, the system owners attended the in-house software course. On completion of this contract, the new software system provided scenario calculations that could take less than one hour to compute. Calculating the number of scenarios normally undertaken each year and the time saving gain from the new system it still took three years to provide a break-even point and a return-on-investment for its procurement. However, the responsiveness of the calculations meant that optimum maintenance and production schedules could be developed in the same responsive manner. The benefit associated with the new faster and more accurate scheduling of maintenance and production activities is very significant as this is a high resource area.

This case study also provided a means to test the use of staff turnover for calculating the benefit from training. It was calculated that the time taken to learn a system with good documentation was estimated to be approximately two-thirds of the time needed to learn about a system without any documentation. The system administrator and developer had estimated that it took nine months to become proficient with the system without documentation. The output from this learning period was, in the words of the developer, "the documents I really needed". An opportunity arose for a new graduate engineer to assume the role of administrator and developer. The time taken for this graduate to become proficient and implement improvements was just three months.

Another case study for the benefits of applied learning was in the use of some scientific analysis software. The development of a local SQMS in response to the knowledge and skills learned on the in-house course meant that the creation and review of important quality documents, such as the project management plan, took four weeks. In deciding the development approach or life cycle as part of the project management plan, it was noted that the software evolution was responsive to each customer request in turn. Each request could take four weeks to develop that customer solution. As a direct result of the training on appropriate life-cycle selection and implementation of quality review procedures to create a project management

plan, the project managers sought resource to facilitate a demonstration of current software capability to all prospective customers. The conclusion of this exercise was that requirements were requested in one well managed meeting that reduced the development burden considerably. It was estimated that 12 weeks were saved in development time. So the four week documentation schedule and decision making captured in the management plan saved eight weeks from the schedule.

In both the case studies described, the benefits of using best-practice software engineering techniques and in implementing the SQMS can be directly attributed to the training provided. Without this training there is no reason to suggest that the project managers would have changed their approach and would continued as they had before. The self-assessments that the SQMS was poorly implemented before the training, indicating that the improved methods used after training were as a result of what had been learned on the course.

8.5 Analysis

Table 8.2 provides a powerful example of the benefits in providing internally developed and managed training courses in comparison with costs associated with attending external courses. The nine courses were run in the three years, 2003 to 2005, so this took 27 trainers days plus a total of 18 days for administration and course preparation. Much of the preparation was associated with case study material and the self-assessment that was sent out prior to the course. The results demonstrate very effective resourcing as the cost of the time to provide the training was approximately £9,000 and therefore the total cost saving, £132,500.

The use of self-assessment techniques to assess applied learning and evaluate associated efficiencies is also very powerful and cost effective. It does emphasise the need to have self-assessment as a pre-requisite for the training in order to provide a

baseline from which to measure any improvement resulting from the training. This then enables the measurement of the applied learning back on the job, which is the true objective of any course.

Table 8.5 Training Course Results

Issue	Year	2000	2001	2002	2003	2004	2005
Average course evaluation		1.78	2.13	n/a	2.01	1.9	1.77
Average test result		n/a	n/a	n/a	64.3%	71.4%	77.7%

Key: A course evaluation score of 1 is excellent, 6 is very poor.

The training course evaluations shown in Table 8.5 have produced excellent results. The core issues considered more important, such as meeting course objectives and trainer knowledge, scored particularly well, though other aspects, such as joining instructions, food and building environment issues scored less well. The initial course of 2000 was marketed as a pilot course requiring endorsement to continue by the delegates. The novelty of this pilot may account for the very good initial score of an average 1.78. The course tests were introduced because some of the case study work was not being performed accurately every time. The tests were made challenging to ascertain the true level of the learning taking place. It also gave feedback that allowed the trainer to modify or increase emphasis on learning points. Predictably the results improved each year as training course content was modified to take-on the extra emphasis.

8.6 Conclusions

The methodology presented and associated results show that the training provided and learning experience created has been successful in demonstrating the benefits that training can have in the workplace. This success can be quantified by the cost-benefits and the financial return-on-investment results.

The difficulties associated with demonstrating cost-benefits have also been conveyed, particularly the accuracy of recording activity duration. The human perception factors and the reticence to convey efficiencies can be significant hurdles to overcome.

Demonstrating a cost-benefit is a powerful means by which any quality manager can influence working practices. Overcoming the perception that quality assurance can be a bureaucratic overhead is a significant challenge, so any methodology to demonstrate its value can be highly influential

8.8 Recommendations

There are a number of improvements that have been identified to further improve the training and learning experience currently provided. To further assess the learning actually gained from the training and, perhaps more importantly, identify delegates with less knowledge, a test could be conducted at the start of the course. This could be a simple questionnaire to ascertain delegate's confidence on relevant topics. Also, some of the effective learning methods presented in the literature review could be utilised for course improvement.

If the use of self-assessment as a pre-course requisite proves troublesome, the development of action plans, in-line with Kirkpatrick's (1998) level 1, "planned action", could be incorporated into the course.

More return-on-investment case studies could be used to build-up expertise to improve the training courses, and to act as a powerful and influential marketing tool to promote the training courses and the software quality management system by demonstrating their value and benefits.

Providing demonstrable return-on-investment for the training has increased the understanding of the benefits of training for management which has enabled better planning and budgeting for the courses. The methods used have also enabled the courses to be continually improved and the benefits of the software quality management system to be demonstrated.

Since the initial paper was written the course has been further developed to include a pre-course questionnaire that has highlighted the knowledge gained by delegates. Also, the number of PowerPoint slides was reduced and more hands-on exercises introduced to coincidence with the concentration cycles and deal with the early afternoon "graveyard shift".

Chapter 9. An Evolutionary Cultural-Change Approach to Successful Software Process Improvement

Chapter Preface

With a plethora of models, systems and standards to choose from as basis of software process improvement, decisions on which to adopt may depend on a number of factors. This chapter presents an evolutionary and extremely cost effective approach to implementing a software quality system that requires minimum resource and little disruption to programme delivery.

This chapter is based on "An Evolutionary Cultural-Change Approach to Successful Software Process Improvement" (Elliott et al, 2007b) in the proceedings of the Software Quality Management (SQM) conference in April 2007.

The method presented, achieved a 40% improvement in the level of implementation of the AWE plc software quality management system over a five year period. A critical success factor is the treatment of the users of the defined software quality system as customers, understanding their concerns and problems, and being responsive to them. The importance of a well designed system is highlighted together with the essential and extensive consultation process required to gain buy-in and lay the foundation for cultural change. This was supported with a helpful programme of facilitated self-assessment and sustained by a closely aligned training scheme. As a consequence some of the cultural elements were changed from one of thoughtless "tick-in-the-box" compliance to one of true understanding of the system requirements, true quality implementation, and subsequent added value.

9.1 Introduction

This chapter documents a successful methodology for improving the level of implementation of the software quality management system (SQMS) at AWE plc.

The success of the methodology can be presented in three ways; firstly by direct measurement demonstrating the improved level in implementation, secondly, by an examination of the reduction in resources needed to achieve this improved level as compared with that used previously, and lastly by the cultural change that took place highlighted by the differences in behaviour before and after, or indeed during the achievement.

The methodology improved the level of implementation of the SQMS at AWE from a baseline value of 34% to a measured 74%. These figures are the average values for each year from a number of assessments conducted on software systems throughout the Company. These assessments took place over a six-year period, from 2000 to 2005. The Company quality programme was initially established in 1988 in preparation for possible BS 5750 (as it was then) standard certification at some later unspecified date. The achievement of ISO 9001 certification became the Company's highest priority from 1993 to 1994 as it changed from being part of the Ministry of Defence Procurement Executive to become a contractor operated establishment. The requirements of the SQMS had to take into account both the ISO 9001 standard, utilising the supporting guidance standard ISO 9000-3 and the Defence Standard 05-95, Quality System Requirements for the Design, Development, Supply and Maintenance of Software. At its peak, five full-time staff were assigned to the SQMS definition and implementation in a central quality organisation. To help establish Software Quality Assurance (SQA) principles, a further 70 co-ordinators, known as local area software representatives (LASRs) were appointed. In addition, the central software quality group responsible for defining and implementing the SQMS at AWE, had at its disposal the internal audit team to check, or perhaps more aptly, enforce compliance. Arguably this resource had achieved the aforementioned 34% level of implementation.

The increase to the measured improvement of 74% was achieved with just the one full time software quality manager, the author. Of particular interest was the significant difference after the application of the methodology described in this chapter that

occurred in behaviour from software practitioners, the customers of the SQMS. At the initiation of the improvement project, some components of the culture could be described as "thoughtless tick-in-box compliance". Some of the outputs from various processes mandated or documents produced in response to the SQMS requirements could lack quality. A detailed review of many of these documents would conclude that some were produced; "just to get them out of the way", indeed this was admitted by many of the document or software system owners. From earlier research (Elliott et al, 2006b) discussed in Chapter 6, it was evident that there was a level of resentment in having to implement what were not fully understood controls. The success of the improvement programme was to provide software developers with a good understanding of the SQMS requirements, and to encourage active engagement with these customers to deal with their concerns. As a result, more quality documents were produced that contained the output of good decision making that improved both the effectiveness and efficiencies of local working practices. It could therefore be demonstrated that implementing the requirements of the SQMS truly added value and this could be further emphasised by well maintained systems.

9.2 Methodology

A project to improve the adoption of the SQMS was initiated at the completion of a comprehensive assessment on the level of implementation undertaken in 1999 (Elliott et al, 2006b) that used the assessment question set presented in Annex 1 and discussed in Chapter 6. This comprehensive assessment, in fact, became the first step of a seven-step methodology for effective implementation of an SQMS developed at AWE plc. Some of these steps were conducted in parallel, such as the user feedback being gathered during the establishment of the baseline. Also training delegates could conduct self-assessments as part of their training and then be further supported in subsequent implementation of their learning. This is depicted in Figure 9.1.

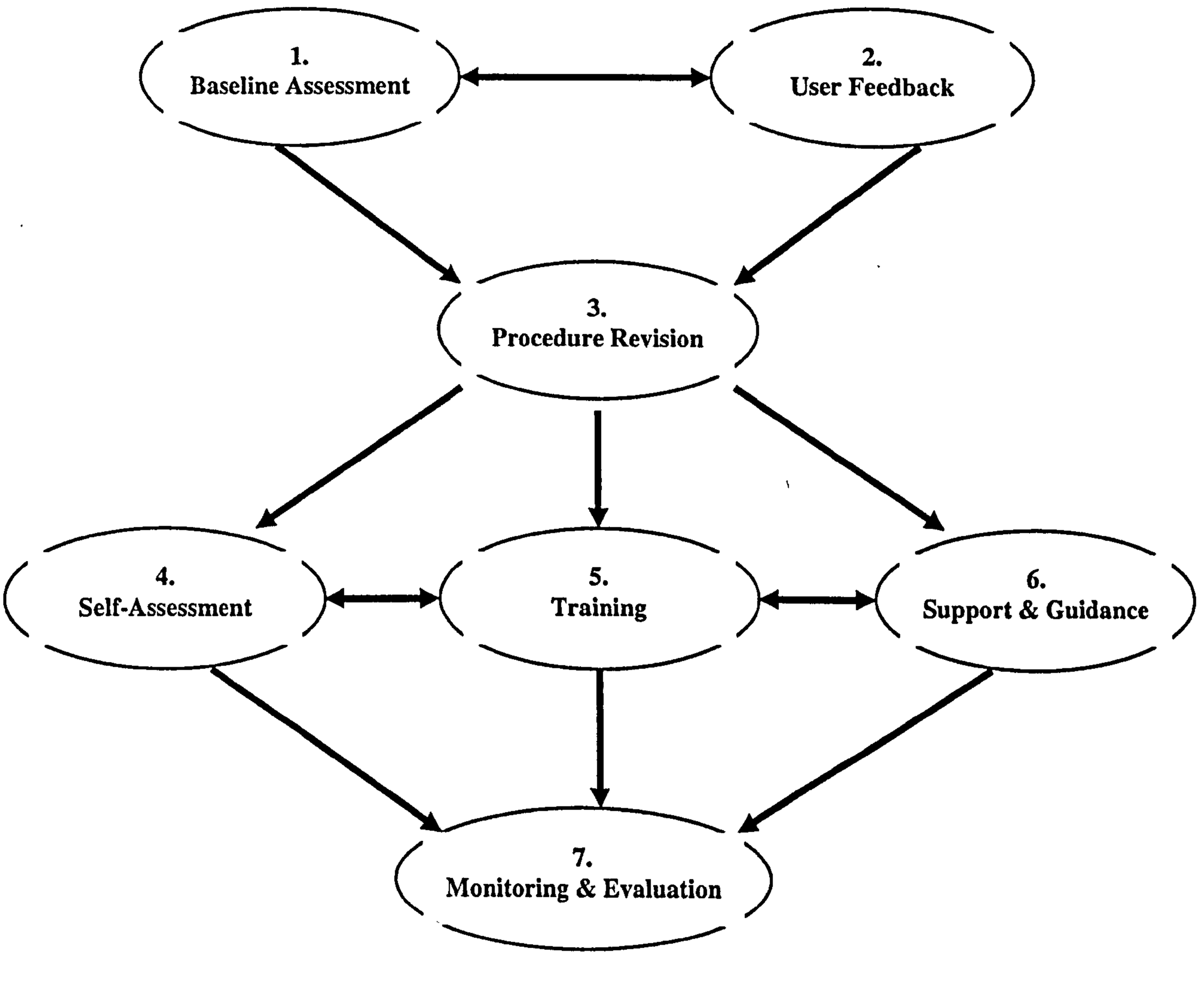

Figure 9.1 Representation of the seven step methodology.

Step 1 : Establishing a Baseline

A sample of fifty-five software systems from a diverse range of applications, were assessed against all the requirements of the SQMS. These requirements were documented in a comprehensive audit question-set that were grouped into software quality topics. These included general management, requirements management, development, testing, use, configuration management, and disaster recovery.

A range of 0 to 4 implementation rating system was applied to each topic. This rating system could be described as an anti-dote to the culture of "tick-in-the-box" compliance as level 4 was a test that the local area valued their implementation of the SQMS. Increased levels on the rating for a topic could be achieved with an increase in the number of requirements for that topic that were complied with. This has similarities to the Software Process Improvement Capability Determination (SPICE) model of ISO Standard 15504 (ISO, 1998). However, level 4 could not be achieved unless the initial actions to implement the SQMS were updated when changes occurred, so that the system retained its integrity and was maintained. This would act as a demonstration that the local area felt what had been implemented was of sufficient value to be maintained. This provided an indication that SQMS requirements were an established way of working and had become institutionalised, and not completed and then forgotten. Each system assessment attracted a pro-rata percentage of implementation that was collated to calculate a company average for the year (Elliott et al, 2006b). The result came to 34% for 1999, and this was a concern for senior management. Of particular concern was that neither the ISO 9001 assessments for certification nor the internal audit process had raised any major issues. It was directed that corrective action for improvement had to be conducted in a low profile manner until significant improvement could be demonstrated. A programme of assessments continued each year as can be seen in Table 9.1. Unfortunately, due to a temporary reassignment on to other tasks for most of 2002, only five assessments took place on software systems and these were all well managed.

Step 2 : Gathering Feedback from Users

In parallel with Step 1, the assessment of 1999, was the gathering of comments, problems and other barriers to adopting the SQMS, and integrating the SQMS with the way employees worked. These issues were documented as "causes" (eg direct, contributory, root) for non-compliance on the assessment pro-forma. There were clear themes presented in these comments. There was certainly a significant cultural resistance to the procedures which, in many cases, stemmed back to the original ISO

9001 certification process in 1994 when many practitioners felt that their concerns on the system definition were not taken into consideration. As a result, a culture of annoyance and an assumption of lack of relevance was evident concerning the SQMS. This would present a significant challenge to improvement. From a technical viewpoint it was clear that some terminology was not understood, and was known as "quality speak". There was criticism that all the procedures had to be read before you knew what rules to follow. This was often described as an inability to visualise the various processes. There was a lack of understanding of many software quality and engineering concepts. Best-practice requirements management, life-cycle selection, configuration management and effective reviews were not fully understood.

Step 3 : Review and Revision of Quality Procedures

To address the lack of clarity of the procedures and deal with some of the other comments or criticism of the SQMS, all the procedures were reviewed and revised. Based on the feedback from Step 2, key considerations for the revision were to deal with the lack of coherence of the current system, visualisation of processes and the provision a simple framework or introduction to the main requirements. The solution to the issue of applicability was to develop a system to grade or tailor the number of requirements based on risk and complexity. The re-drafted procedures formed the basis of a significant consultation process. The drafts were sent out for comment, and were facilitated by a number of presentations and workshops organised by the author. Over thirty presentations were held by the author, with audiences ranging from four or five to around thirty. Whilst most of the events were pleasant, some became quite heated. The aims of this consultation were marketing the benefits of implementing procedures and dealing with any concerns so as to gain the buy-in from the practitioners, the users of the system. It was hoped that this would help overcome much of the resistance that had built-up over the years. These workshops and presentations certainly raised the awareness of the procedures, but it became evident that education on software engineering, quality principles and terminology conversion was required. This whole process took four months, however, the improvement in

understanding and reduction in resistance meant this time was well spent and ultimately quite effective.

Step 4 : Facilitated Self-Assessment

The next phase of the improvement programme was to carry-out a series of facilitated self-assessments. This would continue the non-threatening and low profile approach established from the consultation process. Also, when advice and guidance was requested from the central quality group by software developers, they were encouraged to conduct a self-assessment as described in Chapter 7. However, it was not mandated to engage in a process improvement project. In addition, developers were encouraged to attend a training course on software engineering principles and quality as described in Chapter 8. Self-assessment was particularly useful in support of the training as not only did it baseline the current level of implementation, but it helped gain an insight as to the amount of understanding of software engineering or quality processes. This helped tailor the training to specific individual training needs and present examples that were more relevant to attendees.

Step 5 : Software Engineering Training

Step 5 was carried out in parallel, and working closely with Step 4. The first training course was started in late 2000. The philosophy for training was to provide a good introduction to the principles of software engineering and link this into how to implement the company software procedures. As described in Chapter 8, a partnership was established with a training provider that not only had a ready made software engineering course, but also had experience of implementing the methods and techniques from a diverse range of software systems similar to AWE. The author contributed to the training by providing the links to the Company Software Procedures.

From the onset it was felt important to ensure attendees felt the course was of value. This was not only achieved through the monitoring of course evaluations but also by comments requested at the end of each course tutorial. The evaluations consisted of a

comprehensive set of questions on course objectives, value, applicability, joining instructions, food, etc., to be answered on a six point scale ranging from 1 for excellent to 6 for very poor. After a number of courses it became clear that the AWE specific elements was considered the most valuable, so the course altered to become a completely bespoke, tailor-made to the company software system. Nearly all tutorials were geared to further implementing the SQMS requirements. Attendees would document there own processes, collate inventories and assign categories and software product baselines. They also learned to understand system measures and metrics and apply their own. The courses were not made compulsory but the skills provided by the course were documented in a competency framework within the company software procedures. Over a six year period, from 2000 to 2005, 130 people completed the training course, and a total 69 full assessments were conducted as part of the training.

Step 6 : Providing Support and Guidance

Step 6 was also undertaken in parallel with Steps 4 and 5. Another issue that needed solving that became apparent during the consultation process of 2000, was the considerable time spent explaining many aspects of the software quality system on the telephone. Frequent questions were on some quality terminology and how to implement various requirements in a range of differing situations, and there were many requests for examples. These questions continued after the procedures were published. The role of central quality assurance could be compared to that of a helpdesk. Unfortunately many of these calls would last typically 20 minutes with a significant number around 45 minutes. The response to this required a significant resource and presented a clear need to support or underpin the main procedures with guidance information. Certainly the ability to refer, in the first instance, to a document would alleviate the amount of time spent of the phone. In response to this need, a programme was established to produce guidance documents. To-date, 22 guidance documents have been issued to provide the detailed explanation, examples and templates to streamline software engineering document production. The first document produced was a template on how to produce a software control plan. As

this would provide the biggest return-on-investment and immediately improve both the quality of the processes and significantly raise the level of compliance. Thereafter, the guidance was produced on a priority basis that reflected the weakest implemented processes, such as software configuration management and guidance for conducting reviews, and the most frequently asked questions, such as "what is the category of my software?" and "what is the difference between verification and validation?" These documents were published and made available on the company intranet

Step 7 : Monitoring and Evaluation

The final step is closely associated with Step 1, Establishing a Baseline. The exercise to assess the level of implementation of the SQMS was continually monitored with each new quality assessment, with the level of compliance with the SQMS being recorded. At the end of each year the results were compiled into a table as shown in Tables 9.1 and 9.2, to gain a measure of what improvement in implementation was being achieved, which then gave a measure of the success of the overall improvement methodology being implemented. The results of this monitoring are given in Section 5 and these are discussed in Section 6.

In practice, all the above steps were subject to many iterations. Each self-assessment helped identify areas of improvement in the training, in the support and guidance and even in the quality procedures themselves. The questions trainees asked in training sessions also led to improvements in support documentation and the greater understanding the training imparted led to improved self-assessments. The questions received in the support process identified areas of training need and the training courses and support contacts helped recruit further candidates to carry out self assessments. This continual improvement in all of the steps in the methodology was an essential part of the approach, allowing the improvement process itself to grow in its effectiveness as it was being carried out and developed.

9.3 Results

Table 9.1 Software System Assessment Results

Year	Number of Systems Assessed	Average
1999	55	34%
2000	18	46%
2001	19	54%
2002	5	83%
2003	22	60%
2004	21	66%
2005	16	74%

Table 9.2 Software Quality Topic Percentage Improvements by Year

Topic / Year	1999	2000	2001	2002	2003	2004	2005
General Management	35.38	51.1	56.83	n/a	60.6	64.59	71.88
Requirements Management	26.5	32.7	41.5	n/a	57.30	70.53	76.25
Development	26.5	44.4	48	n/a	58.88	60.47	68.93
Testing Arrangements	40	53.2	60.17	n/a	68	70.71	75.5
System Use	n/a	58.9	69.50	n/a	74.4	75.47	94.15
Configuration Management	19.5	26	32.56	n/a	49.50	52.24	67.44
Disaster Recovery	35.75	47.1	60.72	n/a	58.3	57.36	74.06

Table 9.3 Training Course Results

Issue / Year	2000	2001	2002	2003	2004	2005
Number of Attendees	14	23	n/a	30	30	33
Average Course Evaluation	1.78	2.13	n/a	2.01	1.9	1.77
Average Test Result	n/a	n/a	n/a	64.3%	71.4%	77.7%

Key: A course evaluation score of 1 is excellent, 6 is very poor.

9.4 Analysis of Results

The results in Table 9.1 show a steady improvement in the overall level of implementation of around 7% each year apart from 2002. During 2002 only five systems were assessed which were all well managed and rather skewed the trend. As well as the increase in the yearly average, the number of systems below 40% reduced from 27 in 1999 to just the one in 2005. Although there was a programme of

assessments a significant number of systems assessed each year came from either people requesting advice or from attending the training course. In this respect the sampling of assessments contained an element of randomness adding to the validity of the improvement.

Another indication of the general upward trend is that a number of systems were assessed more than once and on each occasion an improvement was achieved. The owners were therefore implementing more activities to implement the SQMS through their own initiative. This is further indication that once understood SQMS requirements are seen as adding value.

The topic improvements in Table 9.2 provide a good indication that the training emphasised on both configuration management and requirements management have had significant impact as they are the topics with highest improvement. The improvement in development is in the area of deciding the design needs based on reliability and maintainability. The significant increase in the "system use" section revolves around the fact that people can see an immediate payback to implementing these actions or find it easier to see the relevance in adding these controls. Within the topic areas the activities least undertaken or not undertaken well are:

- Quality reviews with too much reliance on signature authorisation.
- Maintaining an "as is" requirements specification throughout the entire life cycle of the system.
- Good quality in design reviews.
- Test planning and implementation to demonstrate code coverage and boundary value analysis.
- Configuration management planning to develop a hierarchy of baselined configuration items.
- The development of product and process measures or metrics to demonstrate quality.
- The testing of a disaster recovery plan.

assessments a significant number of systems assessed each year came from either people requesting advice or from attending the training course. In this respect the sampling of assessments contained an element of randomness adding to the validity of the improvement.

Another indication of the general upward trend is that a number of systems were assessed more than once and on each occasion an improvement was achieved. The owners were therefore implementing more activities to implement the SQMS through their own initiative. This is further indication that once understood SQMS requirements are seen as adding value.

The topic improvements in Table 9.2 provide a good indication that the training emphasised on both configuration management and requirements management have had significant impact as they are the topics with highest improvement. The improvement in development is in the area of deciding the design needs based on reliability and maintainability. The significant increase in the "system use" section revolves around the fact that people can see an immediate payback to implementing these actions or find it easier to see the relevance in adding these controls. Within the topic areas the activities least undertaken or not undertaken well are:

- Quality reviews with too much reliance on signature authorisation.
- Maintaining an "as is" requirements specification throughout the entire life cycle of the system.
- Good quality in design reviews.
- Test planning and implementation to demonstrate code coverage and boundary value analysis.
- Configuration management planning to develop a hierarchy of baselined configuration items.
- The development of product and process measures or metrics to demonstrate quality.
- The testing of a disaster recovery plan.

The training course evaluations shown in Table 9.3 have produced excellent results. The core issues considered more important, such as meeting course objectives and trainer knowledge, scored particularly well, though other aspects, such as joining instructions, food and building environment issues scored less well. The initial course of 2000 was marketed as a pilot course requiring endorsement to continue by the delegates. This may account for the very good initial score of an average 1.78. The course tests were introduced because some of the case study work was not being performed accurately every time. The tests were made challenging to ascertain the true level of learning taking place. It also gave feedback that allowed the trainer to modify or increase emphasise on learning points. Predictably the results improved each year as training course content is modified to take-on the extra emphasis.

9.5 Conclusions

It is clear from these data that steady improvement in implementing the software quality system has occurred year on year and that the overall approach in facilitating these improvements has been successful and effective. However, it could be questioned whether, over the six year duration, the approach could have been made more efficient. As the approach showed initial success, a business case could have been developed to increase the resource and support for the initiative and perhaps accelerate the programme. However, it should be remembered that the initial programme had to be carried-out in a low profile manner without senior management support until a reasonable level of improvement could be demonstrated. Also, during 2002 the programme of work was placed on hold, as other tasks took priority. Indeed there was some doubt at that time about whether the project could be continued. So it is perhaps fair not to underestimate the impact this had on the programme in terms of lost momentum. Perhaps, now in times of stability, a business case to present to senior management could be reconsidered, but this is beyond the scope of this thesis.

An independent validation of the improvement is given by the ISO 9001 Certification process where the number of problems found in software issues has reduced. Only one, minor nonconformity has been raised in the last three years and only five actions requiring correction. Perhaps the most pleasing aspect of the improvement is that it reflects a difficult to achieve success for software process improvement projects that this approach has truly won the hearts and minds of practitioners to bring about a change in culture and this success has led to sustainable improvement. With a philosophy of customer care, consultation, and active engagement, practitioners now produce documentation of good quality that, in turn, facilitates good decisions that have proved to be effective and efficient.

9.6 Recommendations

One problem with the methods described in this chapter is that a number of people did not engage in improvement projects. To overcome this problem and guarantee engagement, the self-assessment could form part of an audit programme. Support and guidance would still be provided as would the training course. Certainly things could be accelerated if the cost-benefits could be clearly demonstrated and applied to a business case that requested and achieved additional resource. Also, improvements to the training programme could be considered or indeed extended into other closely related topics, providing more details of good software engineering practice. A specific training course for conducting self-assessments could be also considered. However, the time taken to implement these improvements means it is not possible to try these improvements within the scope of this thesis.

The approach outlined by this chapter has achieved a unique a very successful outcome that overcame cultural resistance and changed working practices. The cost effectiveness of this methodology means its adoption could be considered by any organisation.

Chapter 10. Implementation - Testing the general applicability and value of the research findings

Chapter Preface

This chapter provides a further verification of the successes of the methodology as some of the recommendations from early chapters are tried and tested. The recommendation of Chapters 4 and 5 on the internal audit process were actioned with results discussed. The chapter finishes with a discussion on the use of self-assessment as a mandatory requirement of an audit together with piloting the assignment of costs to non-conformance utilising financial models of efficiencies assigned against software quality management activities.

10.1 Introduction: Audit improvements

The final stage of the research was to implement some of the recommendations from the earlier chapters. This again formed quite an integrated approach as the recommendations from the auditing Chapters 4 and 5 were combined on a new audit programme mandating the self-assessment spreadsheet as a check-list as suggested in Chapters 7 and 9. Further, the cost models created in Chapter 8 were also included in the self-assessment spreadsheet to test the response of financial factors being applied to inefficiencies related to inadequate practice, or in auditing terms, non-conformance. A follow-up questionnaire, similar to the one used in Chapter 5, was used to test improvements in the approach.

10.2 Research implementation objectives

The research implementation objectives of this final phase were to utilise the AWE plc Company Audit Programme to test the following action suggestions resulting from the research findings of previous chapters:

- utilise a comprehensive check-list to test entire system implementation
- create a management system customer who "cares" about the audit findings
- mandate self-assessment as an auditee task
- facilitate root cause analysis and risk analysis for non-conformance
- introduce financial models for non-conformance, to test the response of auditees and senior management and seek refinement of the models
- provide direct support for necessary process improvements by audit deficiencies.

10.3 Method

The management areas chosen for the audits were selected on the basis that they had not engaged on improvement initiatives over the last few years or had not undertaken the self-assessment as part of the training programme as described in Section 10.2. The selected areas were placed on the company audit programme which is published quarterly. One month prior to audit, the local area would be notified of the impending audit. This would have a wide circulation to include senior management to try and facilitate management support. The notification would comprise an email with attachments of the formal audit notification and the self-assessment spreadsheet. The

email would state that the completion of the self-assessment was mandatory, however help would be provided in the case of not understanding the interpretation of some of the questions. The point of contact for the audit would also be telephoned to allay any concerns and to start the support and engagement process. The points of contact were encouraged to complete the self-assessment as audit preparation prior to the audit investigation and forward it for review. The audit investigation path would be similar to that described in Chapter 4, the local inventory of software items would be reviewed first followed by the main document that describes local working practices known as the software management and control plan. As a good understanding on local practice was developed the self-assessment questions would be further reviewed to check detailed implementation. Audit deficiencies raised as non conformances uncovered, were presented to senior management to ensure support and drive for improvement actions. Support to improve local practices such as document reviews and advice was provided until all audit actions were completed and the relevant audit documentation completed. The results recorded were the initial assessment result and the percentage improvement as shown in Table 10.2.

Four weeks after completion of all audit actions, a questionnaire was sent out, from the audit department, to test auditees' responses to the audit process. These results are presented in Table 10.3.

10.4 Results and Analysis

Software quality management system assessment results for 29 (2006) and 51 (2007) software systems across AWE plc are presented in Table 10.1. The detailed results for each system are collated in Annex IV.

Table 10.1 Audit Assessment Results for 2006 & 2007

Year	Number of Systems Assessed	Average Percentage Implemented
2006	29	73%
2007	51	81%

10.4.1 Software quality implementation assessment results

There were many different types of software systems assessed across AWE plc from a few simple spreadsheets manipulating important company data to vast modelling software systems calculating weapon characteristics. Each functional area was visited. The value of the percentage implemented is based on average final assessment results from the 29 systems assessed in 2006 and 51 systems assessed in 2007. Improvements in the implemented software quality system can by highlighted by only five assessment results that were below 50% in 2006, highlighted in bold font in Table IV.1, compared to 47 in 1999. The assessments were conducted in management areas that had no contact or involvement with the software quality management initiatives over the years 2000 to 2005. The bold systems that appear together will usually be in the same management area, reaching similar results. This was the case in both 2006 and 2007. It was reassuring that management areas visited that had attended the training course but had not undertaken self-assessment had reasonable scores, despite concerns to the contrary. The company average score of 73% for the year is a good reflection on the progress for the process improvement actions identified from both the audit and self-assessment.

It was noticeable that the drive to complete audit actions which are tracked on a company data base provided more impetus to improvement action than normally witnessed through facilitated self-assessment without audit. Table A IV.2 shows that the initial assessments for 2007 were higher than 2006 and more in line with trends from previous years, as depicted in Table 9.1 in Chapter 9. Four systems were below 50% in 2007 but, considering the number of systems assessed was high, 51, this can be seen as positive. Again the drive given to progress audit action accelerates the audit and self-assessment process improvement activity. The increase to 81% is again in line with the trends from Table 9.2 of Chapter 9.

Table 10.2 Assessment results for key software quality management system topics

Year	Use	Testing	Req'ments	Disaster Recovery	Development	General Man'ment	Config Man'ment
1999	no data	40.00%	26.50%	35.75%	26.50%	35.38%	19.50%
2000	58.90%	53.20%	44.40%	47.10%	44.40%	51.10%	26.00%
2001	69.50%	60.17%	48.00%	60.72%	53.20%	56.83%	32.56%
2002	no data	no data	no data	no data	no data	no data	no data
2003	74.40%	68%	57.30%	58.30%	58.88%	60.60%	44.44%
2004	80.00%	70.71%	71.44%	57.06%	60.47%	64.59%	57.77%
2005	89.15%	75.50%	75.00%	74.06%	68.93%	71.88%	63.30%
2006	78.10%	78.51%	81.72%	78.88%	71.11%	78.15%	44.50%
2007	84.04%	80.76%	83.30%	84.77%	79.70%	81.45%	73.00%

10.4.2 Software quality topic data:

It should be noticed from Table 10.2 that the results for the main software quality topic areas start to converge towards 80%, with configuration management (CM) being the exception. One of the reasons for the delayed improvement figure for CM is that hierarchies of baselines are often last to be completed. These baselines could have some configuration items (CIs) missing that will be developed when an associated change occurs to warrant the production of the CI. All these will lower the percentage implementation for configuration management. Configuration

management implementation had a low initial baseline measure so more improvement action was required. It is also noted that there is a reticence to conduct audits on the configured system and this can also have a significant impact on the results.

Another area of reticence is the development of process or product measures and metrics. In the spirit of continuous improvement utilising data captured on either process efficiencies or product quality is strongly encouraged. Fear of micro-management or demonstrating inefficiencies in their working practice or arbitrary targets being set, tends to dominate behaviours. The area where data is frequently gathered is the capturing of the number of problem reports. These could be the number of faults for each main version release and dealing with these faults on a case by case basis, looking for trends or repeated incidents.

There is no data for the analysis in 2002 as the author was assigned other tasks away from the software quality system for that year.

10.4.3 Audit questionnaire results

A significant difference between the 2003 results and 2007 is that the variation in scores is low. The exceptions are the conducting of root cause analysis and the level of auditor understanding. The high variation exhibited on the root cause analysis question is because some auditees were more methodical when undertaking the self-assessment. Others neglected or choose to ignore the chance to analyse the underlying causes. The slight increased variation on auditor knowledge is centred around the accuracy of the question in that it is open to misinterpretation whether the question relates to the auditees' work or software quality management. Auditees responding with lower values tended to relate the question more to their own work. This was uncovered by a sample of ten follow-up interviews. The overall reduction in

variation of the questionnaire responses will be, in-part, due to a single auditor, the author, having a consistent approach and style. A reasonable estimate of the number of auditors whose approach is reflected in the 2003 questionnaire would be as much as fifty, which would bring about an element of variation. However the audit programme of 2007 did investigate an extensive range of technologies, the type of work, the people, their backgrounds and their appreciation of auditing.

Table 10.3, Audit survey questions and results

1. Overall how would you rate the effectiveness of the audit process?

No value (1)	Little Value (2)	Some Value (3)	Moderately Effective (4)	Effective (5)	Very Effective (6)

Auditee Result 2003			Auditee Result 2007		
Average Score	Variation	Answer Range	Average Score	Variation	Answer Range
3.45	High	1-5	4.80	Low	4-6
			Improvement		**39.1%**

2. When you have been audited, how clear were the criteria for a successful audit?

Not at all (1)	Poor (2)	Some understanding (3)	Good (4)	Totally clear (5)

Auditee Result 2003			Auditee Result 2007		
Average Score	Variation	Answer Range	Average Score	Variation	Answer Range
3.05	High	1-5	3.0	Low	2-4
			Improvement		**-0.5%**

3. In general, how would you rate the auditors' understanding of the subject matter in your area of interest?

Auditee Result 2003			Auditee Result 2007		
Average Score	Variation	Answer Range	Average Score	Variation	Answer Range
3.30	High	2-5	4.6	Medium	3-6
			Improvement		**39.4%**

4. How well prepared was the auditors?

Not at all (1)	Little (2)	Some (3)	Prepared (4)	Well prepared (5)

Auditee Result 2003			Auditee Result 2007		
Average Score	Variation	Answer Range	Average Score	Variation	Answer Range
3.45	High	2-5	4.2	Low	4-5
			Improvement		**21.7%**

5. How would you rate the significance of findings?

Totally insignificant (1)	Very minor (2)	Minor (3)	Of some significance (4)	Significant (5)	Very significant (6)

Auditee Result 2003			Auditee Result 2007		
Average Score	Variation	Answer Range	Average Score	Variation	Answer Range
3.60	High	2-5	3.60	Low	3-4
				Improvement	**0%**

6. How much analysis (root cause) of the reasons for the deficiency is undertaken in your area?

None (1)	A little (2)	Some (3)	Reasonable (4)	A lot (5)

Auditee Result 2003			Auditee Result 2007		
Average Score	Variation	Answer Range	Average Score	Variation	Answer Range
3.05	High	2-5	2.8	High	1-4
				Improvement	**-8.2%**

7. How much monitoring occurs to track improvement actions?

None (1)	A little (2)	Some (3)	Reasonable (4)	A lot (5)

Auditee Result 2003			Auditee Result 2007		
Average Score	Variation	Answer Range	Average Score	Variation	Answer Range
2.9	High	2-5	4.2	Low	3-5
				Increase	**44.8%**

8. How much pressure are you under to closeout the deficiency?

None (1)	A little (2)	Some (3)	A lot (4)

Auditee Result 2003			Auditee Result 2007		
Average Score	Variation	Answer Range	Average Score	Variation	Answer Range
2.95	High	2-5	3.4	Low	2-4
				Increase	**15.3%**

9. How would you rate the quality of closeout for deficiencies?

Very Poor (1)	Poor (2)	Reasonable(3)	Good (4)	Very good (5)	Excellent (6)

Auditee Result 2003			Auditee Result 2007		
Average Score	Variation	Answer Range	Average Score	Variation	Answer Range
3.3	High	1-5	4.4	Low	4-5
				Improvement	**33.3%**

Question 1, measuring audit effectiveness is perhaps the pivotal question. The improvement of approximately 40% validates that the changes to the audit process recommended in Chapters 4 & 5 are indeed improvements, as one of the stakeholders, the auditee, is declaring an increase in the value of the exchange compared with that previously found on the 2003 survey.

The results for Question 2 show no real change from 2003 to 2007. This is disappointing as it was declared in the audit notification that the achievement of compliance to the SQMS at the attainment of above 73% implementation, the company average measured by the self-assessment questions, would demonstrate success. An explanation for the low result could be based on the fact that the audit notification and detailed criteria are initiated at the start of the audit whereas the completion of audit corrective action can take several months. As a consequence the declaration of the criteria of a successful audit may well have been forgotten. An improvement action would be to stress at the completion of corrective action the original criteria for success.

The responses to Question 3 were interesting as they reflected some insights into the behaviours of the auditees. As previously stated, the question is not clear on whether the term "subject matter" in the question is software quality management or the particular discipline of the area being audited. It is fair to say that some areas audited have quite complex activities and would require a high level education and years of experience to understand. This presents a challenge for the auditor, particularly when motivated to find an improved way of working that is accepted by the auditee as a true improvement. There is a degree of snobbery exhibited that an auditor will be unable to make an improvement to working practices without the appropriate knowledge and experience. The reality is that the independent view can, and often does, find improvement and that people are often too close to the detail of the work to see

inefficiencies. The variations in responses reflect an unwillingness to accept that someone can spend a few hours analysing working practices and devise improvements.

The improvement of around 20% on auditor preparation shown by Question 4 is based on the overall thought and work behind the audit approach, specifically the use of the self-assessment tool.

The term "significance" in Question 5 is a double edged sword. Although issues of significance could be found, declaring them as significant by the auditee could be construed as a failing by the auditee or the auditee's management and there is therefore a reticence to declare a truthful answer. The purpose of the questions was to ensure that audits are not finding matters of trivia to report on which is one of the criticisms of auditing. The responder bias has rather invalidated the findings. As discussed in Chapter 5, costing the significance of audit findings would alleviate this problem.

The activity to undertake root cause analysis has not improved from 2003 as conveyed by the answer to Question 6. Although, as mentioned earlier, this analysis is prompted in the summary of the self-assessment it really needs to be facilitated at the conclusion and completion of audit corrective action. This is not a well known or practiced technique and provided a further opportunity of value for the auditing process. Providing guidance in the audit process documentation would further support it being undertaken.

The answers to Questions 7 and 8 reveal that that the annoyance associated with the monitoring of actions is high. Indeed when asked for further comments on the question this area was the most criticised. A frequent comment was that auditees felt

there were more people chasing closure than undertaking the improvement activity and there was little offer of help, just concern about close-out. This is a good example of the incorrect behaviours from poorly thought through measures and methodologies. Actions from audit that go beyond their close-out date are monitored at executive board level.

The improvement of 33% demonstrated by the answer to Question 9 from 2003 to 2007 is another pivot result to validate that there are indeed real improvements to the auditing process resulting from changing the process in response to the recommendations of Chapters 5 and 6. The quality of close-out is so key to ensuring lessons are learnt and corrected.

10.5 Conclusion

Overall the results do validate some of the recommendations from Chapters 4 and 5 showing they do bring about an improvement to the internal auditing process. The key results are increases of perception and value and the quality of close-out which are pivotal aspects. Further enhancements could be made in root cause analysis as the results show this needs to be integrated into the close-out documentation and not just mentioned in the self-assessment tool. Further thought needs to be made concerning the accuracy of some of the questions in the survey as, unfortunately, some have been open to misinterpretation.

10.6 Use of financial models discussion

One of the research objectives was to test the Cost-Benefit Audit Methodology (C-BAM) model described and depicted in Figure 5.2 of Chapter 5. The concept was to

include example inefficiencies for non-compliance and have the cost models integrated in the self-assessment spreadsheet. The summary page on the spreadsheet contained the percentage implementation and also an inefficiency cost based on not fully complying with the SQMS. The idea was this would stimulate debate on the model to develop the accuracy and assigned weightings of the costs to the software system and management arrangements under investigation. The cost-benefit of improvements and the resultant savings could then be collated. The response was quite mixed, from genuine interest to develop the cost models to fierce criticism and challenge to the concept of even trying to include it in an audit process, as it was not a company requirement. Of course, this reflects the relative maturity of the management areas to accept criticism of possible managerial inefficiencies. Indeed it was noticed that the further up the management hierarchy the debate went the more challenge there was to the task. Another interesting behaviour was that once the audit investigation was completed and, specifically, the documentation for the definition of audit non-conformance was agreed, the discussions on the financial model ceased. The focus of audit activity was to focus on closure of the non-conformance.

10.7 Financial models conclusion

This pilot scheme to include financial models for cost-benefit analysis was successful as one or two areas engaged in the discussion on how accurate the models were in identifying the cost of non conformance and the associated cost-benefits and savings attributed to improvement actions. The challenge to the approach described in the last section must be noted and the conclusion here is that the request to consider costs of non conformance was too much of a surprise and seen as a threat to local management. Further development of this approach could be better served with a few pilot schemes with willing volunteers. Selling or marketing the concepts would be initiated first, prior to engagement.

10.8 Overall Conclusion

There have been a number of successes to the application of the research methodology as outlined in this chapter. Testing the methods has also identified opportunities for further refinement. The self-assessment tool has proved extremely useful and its adoption can be increased by integrating its use with the management planning cycle which will help the drive to achieve improvements. The feature of using the self-assessment on the audit programme was a success, particularly when filling out the spreadsheet as a face-to-face activity. It made for a more pleasurable experience and provided drive to the improvement. It is now felt that that use of email to drive its use has not proved as useful as it could.

The audit programme provides the authority to conduct a number of system assessments which eliminates the need for cajoling, which is normally required. In this respect, this improves the overall effectiveness of the method as more systems can be assessed and the senior management's drive to complete audit actions improves the efficiency.

The use of the financial model has proved that it has potential but marketing its use needs senior management approval prior to engagement. Further development of these models will take place with the Software Quality Assurance Sub-committee (SQAS) in the USA as part of the UK/US 1958 Technical agreement between the two countries, but this is beyond the scope of this thesis.

Chapter 11. Summary, Conclusions and Recommendations for Further Work

Chapter Preface

This chapter summaries the whole thesis relating the work to the aims and objectives set out in Chapter 1. Conclusions and recommendations for further work are included to provide direction for future research work by the author, AWE plc and interested parties in industry, particularly the Software Quality Assurance Sub-committee (SQAS) to the NNSA/NWC Quality Managers, which is a liaison established for AWE plc with its counter-parts in the USA under the UK/US 1958 Technical Agreement.

11.1 Restatement of Aims and Objectives

This thesis has covered the achievement obtained from an evaluation and use of a range of process improvement methods, tools and techniques. The main aim of the thesis is:

To develop and evaluate a demonstrably effective and efficient software quality management methodology suitable for a technical company such as AWE plc. . To be effective the methodology must deliver an improved conformance to the quality standards adopted by the company, it must deliver real process improvement to the company, and it must be accepted and willingly used by employees at all levels of the company from the engineer developing the software to the most senior management. To be efficient the methodology must deliver a real Return on Investment, adding value to the company within a realistic timeframe (i.e. months and years rather than decades)

The objectives of the thesis were set out in Chapter 1. These are:

1. Conduct a survey of published literature to determine the current state-of-the-art in software quality management with a focus on auditing, self-assessment, training and cost-benefit and to see if there is any best practice methodologies that could be used at AWE plc. This review would also establish the validity of this research and the potential contribution to knowledge that it would make.
2. By inspecting the records of non-conformance raised during audits on software quality management practice, assess the approaches taken in gathering data to identify weaknesses in the process and the impact this has on effectiveness and auditor motivation.
3. Conduct a survey of auditors and auditees and influencing literature, to investigate the reasons and perceptions why audits do not always add value.
4. Develop an improved process model for auditing to overcome the reasons and perceptions why audits do not always add value based on the findings of literature review and survey. This improved process model should be implemented and tested at AWE plc.
5. Determine the extent of practitioner adoption of a software quality management system by devising a measurement system and audit or self-assessment question-set to be utilised on quality assessments at AWE that capture both the level of implementation of the SQMS and the barriers to improvement.
6. Establish a software quality and engineering capability at AWE plc by designing a training methodology from which trainees apply knowledge and skills to improve business practices that demonstrate a cost-benefit and a return-on-investment. This training should be implemented and tested at AWE plc.
7. Design a cohesive and congruent software quality management system having collated detailed concerns and problems from practitioners and facilitated an

extensive consultation process. The design should include a logical process flow and visualisation techniques to communicate the top-level requirements

8. Test the general applicability and value of the research findings by:
   - publishing in peer reviewed academic conference proceedings and journals
   - tracking the level of improvements at AWE plc and the resulting cost-benefit and return-on-investment
   - facilitating its wider use in industry by companies and organisations in the UK and USA.

Various chapters within this thesis have set about trying to prove and test these objectives. The next few sections will conclude how each objective has been tackled and overcome. Some issues or recommendations have been identified and these form the basis to the future research section in this chapter.

11.2 Achievement of Aims and Objectives

11.2.1 Objective 1, a survey of published literature

It was evident from the literature review on audit methodologies that the auditing world was aware of the criticisms that auditing, at times, fails to add value and, as a consequence, is poorly perceived. Despite the assertions in published literature that auditing should add value, the proposed improved auditing model described in Chapter 5, which evaluates the cost of non-conformity to aid the tracking of improvement benefits, was found to be unique. The commonality of best practices activities captured in a range of software quality standards was highlighted. The potential pit-fall to the accuracy of data on self-assessment is well documented, the methodology of facilitated self-assessment, conveyed in Chapter 7, outlines a significant solution. The benefit of evaluating the business impact from acquired

skills on training courses was emphasised in the literature review. The methodology of Chapter 8 is offered as an example.

11.2.2 Objective 2, inspect the records of non-conformance

With the specific focus in understanding problems software practitioners have in applying the rules of a software quality management system, the audit data presented in Chapter 4 was analysed. The analysis questioned not only the data itself but also how the data could have been incorrectly obtained. The research investigation uncovered the people aspect of the audit process from both the path the audit investigation followed and the impact of auditor motivation which identified some serious weakness in the audit process. Valuable lessons were learned to aid further research as documented in to both Chapter 5, on the improvement of the general internal auditing process, and Chapter 6, on the development of a system to find and answer the original research question on the issues practitioners have with adopting a software quality management system.

11.2.3 Objective 3, a survey of auditors and auditees

An invaluable tool presented in Chapter 5 was the survey that captured the perceptions of value that the auditing process had. An extensive range of questions, covering the internal auditing process were prepared and answered. This provided baseline values for the perception of value various process attributes had. This would prove useful later in the research to validate the effectiveness of the recommendations made and met objective 3.

11.2.4 Objective 4, develop an improved process model for auditing

Having identified the main weaknesses in the auditing process, methods could be developed to fix the problem. The subsequent methodology presented in Chapter 5 proved to be unique, as no comparable methodology was found within the literature review. The new model introduces an additional process stage to estimate the costs

attributed to any non compliance found and to use these data to declare a cost-benefit or saving from the resulting improvement action and meets objective 4.

11.2.5 Objective 5, devise a measurement system

The measurement system devised to meet objective 5 was called an implementation rating system. It gathered the data on the software quality management system implementation and placed culture firmly on the agenda. Research investigations determined that many management areas had complied with requirements to purely "tick the box" and not subsequently updated related documents. To "raise the bar" and present a somewhat cultural antidote to this phenomena a series of levels were devised, the highest of which could not be reached for a particular topic area unless there was evidence that compliance was well established and institutionalised as regular practice. This would test whether the local area truly found value in their implementation of the SQMS. The results and comments gathered helped identify the cultural differences between major functional groups but also, and perhaps more importantly, their areas of weakness and this become an effective training needs analysis tool. The results provide a baseline measure from which to track and validate improvement opportunities suggested by the research.

11.2.6 Objective 6, establish a software quality and engineering capability

The corner stone of any process improvement methodology understanding is that if the methodology is not understood it will fail. When compared with world class evaluation frameworks, the objective 6 to establish a software quality and engineering capability methodology followed in Chapter 8 meant that AWE plc was in the top 13% for organisations gathering data on changes in organisational performance (Tennant et al, 2002). A key component of the training methodology was the demonstration of cost-benefit with accurate data on efficiency savings that were aligned to business objectives. It is not surprising then that a significant element of

the success of the research methodology was the establishment of a software quality and engineering capability which met objective 6.

11.2.7 Objective 7, design a software quality management system

By conducting the software quality system as an enabling service, treating the users of the software quality system as customers, regarding them with respect, understanding their concerns and being responsive to them, breaks down the cultural resistance to system adoption. It also provides a feedback loop to system design. Successes captured in Chapter 9 to meet objective 7 convey the benefits of a well designed system that is driven from this customer feedback with validated consultation. Tried and tested techniques of conveying initial top level requirements with which to "draw in" the customer to find out more proved successful. Visualising logical and information flows aids the communication process to customers.

11.2.8 Objective 8, test the general applicability and value of the research

The final phase of the research to meet objective 8 became quite practical in its approach as Chapter 10 portrays. The recommendations from Chapter 4 and 5 on improving some elements of the internal audit process are implemented. Specifically these were related to providing a customer that cared about the audit findings and were motivated to truly correct the root causes of problems found, therefore acting as an agent for change. A follow-up survey was conducted to verify that the internal audit process had truly improved. The initial findings to the use of the C-BAM methodology are communicated in Chapter 5. This was linked with not only the use of the self-assessment spreadsheet, explained in Chapters 6 and 9, but also the financial models created from the Standish factors and cost efficiencies for software maintenance, as conveyed in Chapter 10. There was a mixed response to the costing of inefficiencies for working practices. This varied from a questioning of the details of methodology to a genuine interest in making savings from process improvement. Again this presents further opportunity for additional research.

11.3 Evaluation of Research Methodology

As discussed in the methodology chapter, the research investigation has been heavily influenced by the author's position, as Software Quality Manager, in the Company. A number of approaches have been utilised to analyse their impact or success on software quality management system implementation. The methodology has demonstrated that success is not always guaranteed and indeed a range of strategies is required to change behaviours. At an establishment like AWE plc, differing cultures exist. Where the financial modelling of benefits described in Chapter 8 may impress in a management area, it would not in a science based research project. Here a means to accelerate the acquisition of knowledge and its reliability are key drivers.

The use of case studies and surveys has proved useful as has the ability to follow up any anomalies with interviews. The interpersonal skills used to be responsive to customer needs should not be under-estimated.

11.4 Future Work

There are a number of useful and valuable suggestions for further work relating to this research. Not only is the complete methodology successful in its own right to other scientific and engineering organisations but some of the individual elements of the methodology can be utilised as a focussed improvement initiative.

The recommendations from Chapter 4 are generic to auditing and could be extended to all internal auditing. Specifically:

- A customer for the topic under investigation that truly cares about the result and would be prepared to provide expert advice to improvement activity.
- The use of a checklist to cover an entire system under investigation to ensure isolated issues or problems can be placed in context.

- Use the cost-benefit audit methodology model to eliminate the trivial issues often raised by audit.

The training methodology presented in Chapter 9 was found to an exemplar when compared against best practice evaluation frameworks and industry in general.

- This could be utilised for any training programme to align itself to business objectives and demonstrate its value.
- Training programmes, specifically for the use of the self-assessment spreadsheet could be established.

The initial research on return-on-investment proved that the best returns come from applying the right practice early in the project as opposed to re-work in the maintenance phase. Further research could include:

- Comparison of the success factors, emphasised by the Standish factors (Standish, 2001), being applied to new projects to identify potential weaknesses in the project approach. Test if any omissions resulted in increase schedule, cost or reduction in quality.
- Use the efficiency measures to highlight increased costs for recommended best practices not being applied.

11.5 Conclusion

The research has demonstrated a practical solution to achieving process improvement for software quality management. The cost effectiveness of the approach is attractive to any organisation, as value and a return-on-investment are keys to success. Any system that has a beneficial impact on "the bottom line" will win. A pleasing aspect of the research is that elements of the methodology have also proved successful individually and, as such, each could be applied in their own right.

As this research has surfaced methods that have not been addressed elsewhere by current literature and the practical aspect of the level of achievement have been self validating, the author believes the findings of this research make a significant contribution to the world of process improvement. With the number of process improvement failures still making headlines, it is suggested that companies adopt the suggestions that have emerged from this research to add another dimension to the effectiveness of their process improvement programmes.

References

Arter, D.R. (1989), *Quality Audits for Improved Performance*, ASQC Quality Press, Wisconsin, USA, ISBN 0-87389-057-4

Arter A.R. (2000), *Management Auditing*, Quality Digest, Web site, www.qualitydigest.com/april00/html/management, (visited June 2004)

Baker, N. (2002), *The Value Agenda*, Internal Auditing and Business risk, Institute of Internal Auditing web site, www.iia.org.uk/knowledgecentre/practicecentre.cfm. (Visited August 2006).

Benbasat I.G., (1984), *An Analysis of Research Methodologies.* The Information Systems Research Challenge, McFarlan F.W. (ed) Boston, MA: Harvard Business School Press.

Benbasat I.G., Goldstein D.K., Mead M. (1987), *The Case Research Strategy in Studies of Information Systems,* MIS Quarterly, Vol. 11, No. 3, pp. 369-386

Becker, H. S. and Geer, B (1957), *Participant Observation and Interviewing: A Comparison,* Human Organization 16: pp 39-40. Reprinted in Qualitative Research. Ed W.J. Filstead, (1970), Chicago and Markham

Beckmerhagen, I.A., Berg, H.P., Karapetrovic, S.V. and Willborn, W.O., (2004), *On the Effectiveness of Quality Management System Audits,* The TQM Magazine, Vol. 16, No. 1 pp 14-24, Emerald Group Publishing Limited, ISSN 0954-478X.

Bennett K H, Munro M, Knight C & Xu J (2000). *Informatics Centres of Excellence; Research Institute for Software Evolution.* IEE Computing and Control Engineering Journal 11(4): 179-186.

Boehm, B. et al. (2000), *Software cost estimation with COCOMO II*, Englewood Cliffs, NJ:Prentice-Hall, ISBN 0-13-026692-2.12.

Brooks, Frederick P., (April 1987) *No Silver Bullet: Essence and Accidents of Software Engineering,* Computer, Vol. 20, No. 4 pp. 10-19.

Burrell, G and Morgan, G., (1979) *Sociological Paradigms and Organisational Analysis,* Heinemann Educational Books, London

Carnegie Mellon University, (2001), *Capability Maturity Model Integration*, Software Engineering Institute, Pittsburgh USA.

Cangemi, M. P. and Singleton, T. (2003), *Managing the Audit Function*, John Wiley & sons, Hoboken, New Jersey, USA, ISBN 0-47128-119-0.

Cornford T., and Smithson S,. (1996), *Project Research in Information Systems*, Macmillian Press Ltd, London.

Crosby P.B., (1979), *Quality is free*, McGraw-Hill, New York, USA ISBN 0-451625-85-4

Curtis B, (1990), *The Human Element in Software Quality*, Proceedings of the Monterey Conference on Software Quality, Software Productivity Research, Massachusetts, USA

DoE (2003), Department of Education, Tasmania, web site 2003: Internal Audit, Sampling Techniques, www .education.tas.gov.au/internalaudit/manual/10techniques.htm, (visited October 2004)

Davidson A.R. & Stern L.W. (2004) , *A Quality Self-assessment Model*, Managerial Auditing Journal, Vol. 19, No 7, Emerald, 859-868.

Department of Education (DoE), Tasmania, web site 2003: *Internal Audit, Sampling Techniques,* www .education.tas.gov.au/internalaudit/manual/10techniques.htm, (visited October 2004)

Denscombe, M. (2002), *The Good Research Guide for Small-scale Social Research Projects*, Open University Press, Buckingham

Deming, W. E., (1986), *Out of the Crisis, Quality, Productivity and Competitive Position,* Cambridge University Press, Cambridge, England, ISBN 0-91137-901-0.

Dickson, G. and G. DeSanctis, (1990), *The Management of Information Systems Research*, in Jenkins A..M. (ed), Research Issues In Management Information Systems: Setting An Agenda. William. C. Brown

Dwyer, B., (2001) *Successful training strategies for the twenty-first century*, The International Journal of Educational Management, Vol 15/9 pp 312-318 MCB University Press ISSN 0951-354X.

Fagan, M.E. (1976), *Design and code inspections to reduce errors in program*

development, IBM Systems Journal, 12(7), pp 744-741.

Freeman and Weinberg, (1991), *Handbook of walkthroughs, inspection and technical reviews*, Dorset House Publishing Co Inc.,U.S, ISBN 0º32633196

EFQM (1999), Assessing for Excellence: A Practical Guide fo Self-assessment, European Foundation for Quality Management, Brussels.

Elliott, M., Dawson, R.J. and Edwards, J., (2005), *The Approaches to Internal Auditing - A Software Quality Assurance Case Study'*, Proceedings of Software Quality Management XIII : Current Issues in Software Quality, Bennets, P., Ross, M. and Staples, G. (eds), BCS, SQM2005, Cheltenham, UK, March 2005, pp. 43-55, ISBN: 1-902505-67-0.

Elliott, M., Dawson, R.J. and Edwards, J., (2006a), *Towards real process improvement from internal auditing—A case study*, Software Quality Journal, Volume 14, Number 1, pp 347-363, Springer Netherlands, ISSN 0963-9314.

Elliott, M., Dawson, R.J. and Edwards, J., (2006b), *An Analysis of Software Quality Management at AWE plc*. Proceedings of Software Quality Management XIV : Bennets, P., Ross, M. and Staples, G. (eds), BCS, SQM2006, Cheltenham, UK, April 2006, pp. 17-30, ISBN: 1-902505-76-X

Elliott, M., Dawson, R.J. and Edwards, J., (2007a), *An Analysis of Software Quality Management at AWE plc*. Software Quality Journal, Volume 15, Number 4, pp 347-363, Springer Netherlands, ISSN 0963-9314.

Elliott, M., Dawson, R.J. and Edwards, J., (2007b), *An Evolutionary Cultural-Change Approach to Successful Software Process Improvement*, Proceedings of Software Quality Management XV : Ross, M. and Staples, G. (eds), BCS, SQM2007, Stafford University, UK, April 2007, pp. 1-13, ISBN: 1-902505-76-X

Elliott, M., Dawson, R.J. and Edwards, J., (2007c), *Providing Demonstrable Return-on-Investment for Organisational Learning and Training*, Proceedings of The International Conference of Software Process Improvement Research into Education (InSPIRE) : Ross, M. and Staples, G. (eds), BCS, 2007, Stafford University, UK, April 2007, pp. 130-143, ISBN: 1-902505-76-X

Elliott, M., Dawson, R.J. and Edwards, J., (2007d), *An Improved Process Model for*

Internal Auditing. Managerial Auditing Journal, Volume 22, Numbers 6 and 7, Emerald Group Publishing Limited, ISSN 0268-6902.

Fagan, M.E. (1976), *Design and code inspections to reduce errors in program development*, IBM Systems Journal, 12(7), pp 744-741.

Galin D., (2004), *Software quality assurance, from theory to implementation*, Pearson Education Limited, Essex England, ISBN 0-201-70945-7

Galliers, R. D., ed., (1992), *Information Systems Research: Issues, Methods and Practical Guidelines*, Oxford, Blackwell

Gilb, T. and Graham, D., (1993), *Software Inspections*, Addison-Wesley, Massachusetts, USA, ISBN 0-201631814

Glaser, B., Strauss, A. (1967) *The Discovery of Grounded Theory*, Aldine, Chicago

Glaser, B. G (1992) *Basics of Grounded Theory: Emergence v Forcing* Mill Valley, California, Sociology Press

Guerrero-Cusumano, J., L. and Selen, S., J., (1997), *A comparison of international quality standards: divergence and agreement*, Business Process Management Journal, Volume: 3 Issue: 3 Page: 205 – 217, MCB UP Ltd, ISSN: 1463-7154

Hale, R., (2003), *How training can add real value to business: Part 2*, Industrial and Commercial Training, Vol. 35 No 2 pp 49-52, MCP UP Limited, ISSN 0019-7858

Hirschheim, R.A, (1985) *Information systems epistemology: An historical perspective*, in Mumford et al (1985)

Horch, J.W. (1996), *Practical Guide to Software Quality Management*, Artech House Publisher, Boston, USA, ISBN 0-89006-865-8

Humphrey W.S. (1989), *Managing the Software Process*, Addison-Wesley, Massachusetts, USA, ISBN 0-201180-95-2

Hutchins, G., (2002), *Value Added Auditing*, Quality Digest, October 2002, web site, qualitydigest.com

IAF (2003), *IAF Guidance on the Application of ISO/IEC Guide 62:1996*, International Accreditation Forum.

Iivari, J. (1991), *A paradigmatic analysis of contemporary schools of IS development*, European Journal of Information Systems, Vol. 1, No. 4, pp.

249-272

International Standards Organisation (ISO), (2000), ISO 9001 : 2000 *Quality Management Systems Requirements* International Standards Organisation, Geneva, Switzerland.

IIA, (2004), The Institute of Internal Auditors, web-site, www.iia.org.uk/about/ (visited June 2004).

ISO/IAF, (2003), The Auditing Practices Group (APG) web site 2004: *The role and value of the audit checklist,* www .iso.rog/tc176/ ISO9001AuditingPractices Group (visited October 2004)

ISO, (1996), *ISO/IEC Guide 62 General requirements for bodies operating assessments and certification of quality systems,* International Standards Organisation, Geneva, Switzerland.

ISO 15504, (1998), *Software Process Assessment,* International Standards Organisation, Geneva, Switzerland.

ISO 9001:2000, (2000), *ISO 9001 : 2000 Quality Management Systems – Requirements*, International Standards Organisation, Geneva, Switzerland.

ISO 9001:2004, (2000), ISO 9001 : 2000 *Quality Management System – Guidelines for performance improvements,* International Standards Organisation, Geneva, Switzerland

ISO, (2002), *ISO 19011 Guidelines for quality and/or environmental management systems auditing,* International Standards Organisation, Geneva, Switzerland.

Jones Capers (1994), *Assessment and Control of Software Risks,* Yourdon Press, Prentice Hall Building, Englewood Cliffs, New Jersey, ISBN 0-13-741406-4

Kaplan B., and Duchon D. (1988,) *Combining Qualitative and Quantitative Methods in Information Systems Research: A Case Study*, Management Information Systems Quarterly, Vol. 12, pp.571-586

Karapetrovic, S. and Willborn W, (2000), *Generic Audit of Management Systems: Fundamentals*, Managerial Auditing Journal, Vol. 15 No 6, pp. 279-294, MCB University Press, ISSN 0268-6902.

Karapetrovic S, and Willborn W, (2001), *Audit and Self-assessment in Quality Management: Comparison and Compatibility,* Managerial Auditing Journal,

Vol 16 No 6, pp366-377, MCB University Press, ISSN 0268-6902

Kerstin, S. V., Kleoniki, N.S., and Eleni, V., A., (2006), *Integrating Six Sigma with CMMI.* Proceedings of Software Quality Management XIV : Bennets, P., Ross, M. and Staples, G. (eds), BCS, SQM 2006, Cheltenham, UK, April 2006, pp. 85-95 , ISBN: 1-902505-76-X

Kirkpatrick, D.L. (1998), *Evaluating Training Programs*, Berrett-Koehler Publishers, Inc. San Francisco. CA.

Kolb, D.A. (1984), *Experimental Learning: Experience as the source of learning and development,* Prentice-Hall, Englewood Cliffs, NJ.

Lee, P. and Quazi H. A. (2001), *A Methodology for Developing a Self-assessment Tool to Measure Quality Performance in Organisations*, International Journal of Quality and Reliability Management, Vol 18 No 2 pp 118-141, MCB University Press, 0256-671X

Morey, D. and Frangioso, T. (1998), *Aligning an organization for learning: The six principles of effective learning,* Journal of Knowledge Management, Vol. 1 No 4, pp 308-314, Emerald Publishing, ISSN 1367-3270

Orlikowski, and W.J., Baroudi, J.J., (1991), *Studying Information Technology in Organisations: Research Approaches and Assumptions*, Information Systems Research, Vol. 2., No 1., pp. 1-28

OUP (1996), *Oxford Compact English Dictionary,* Oxford University Press.

Parzinger M., Nath R. & Lemmons M., (2001), *Examining the effect of the transformational leader on software quality,* Software Quality Journal, Issue 9 No 4, December 2001, pp 253-267, ISSN: 0963-9314

Pervan G. P., (1994), *A Case for More Case Study Research in Group Support Systems*, pp.485-496 in TC8 AUS IFIP Conference, Bond University, Gold Coast, Old.

Phillips, J.J. (2002), *Return on investment in Training and Performance Improvement Programs,* 2nd ed., Butterworth-Heinemann, Woburn, MA.

Popper, K., (1959), *The Logic of Scientific Enquiry,* London, Harper

Pronovost, D., (2004) Internal Quality Auditing, ASQC Quality Press, ISBN 08-738947-66

Radice R.A., Harding P.E., Phillips R.W. (1985), *A programming process study*, IBM Systems Journal, Vol 24, No 2 pp 91-101

Rico, D.F. (2002), *Software process improvement: Modelling return-on-investment (ROI)*, Software Engineering Institute (SEI) Software Engineering Process Group Conference (SEPG 2002) Phoenix, Arizona.

Roth, J. (2000), *Best Practices: Value-added Approaches of Four Innovative Auditing Departments*, The Institute of Internal Auditors web-site, www.iia.org.uk, (visited June 2006).

Russell, J.P. (2001), *The Internal Auditing Pocket Guide,* ASQC Quality Audit Division, Quality Press, Wisconsin, USA, ISBN 0-87389-560-6.

Samuelson, P. and Nilson L (2002), *Self-Assessment Practices in Large Organisations, Experiences from using the EFQM Excellence Model*, International Journal of Quality and Reliability Management, Vol 19, No 1, pp 10-23, MCP UP Limited, 0265-671X

Sandi, M. and Robertson, I.T., (1996), *What should training evaluations evaluate?* Journal of European Industrial Training Vol. 20/9 pp 14-20 MCB University Press, ISSN 0309-0590

Sedden, J, (2000), *The Case against ISO 9000,* Oak Tree Press, Cork, Ireland, ISBN 1-86076-0420-2

SEI, (2001), Software Engineering Institute, *CMMI for systems engineering/software engineering/integrated product and process development Version 1.0.* Pittsburgh, PA, Software Engineering Institute, Carneige Mellon University.

Siaksa S, and Georgiadou (2002), *Empirical Measurement of the Effects of Cultural Diversity on Software Quality Management,* Software Quality Journal, Issue 10 No 2, September 2002, pp 169-180, ISSN: 0963-9314

Small, L. (1998), *Respect and How to get it*, Internal Auditor., Aug 1998., P 40-45.

Smith, G. (2005), *Communication Skills are Critical for Internal Auditors*, Managerial Auditing Journal, Vol. 20 No 5, pp. 513-519, Emerald Group Publishing ltd., ISSN 0268-6902.

Standish Group International, (1994), *The Chaos Report*, Standish Group web site, www.standishgroup.com, (visited November 2005)

Standish Group International, (2001), *Extreme Chaos*, Standish Group web site, www.standishgroup.com, (visited November 2005)

Spencer L.M., McClelland D.C., Spencer S.M., (1992), *Competency Assessment Methods, History and State of the Art,* Paper presented at the American Psychological Association, Annual Conference, Boston, USA.

Taormina T (1999), *Successful Internal Auditing To ISO 9000,* Prentice Hall, New Jersey, USA ISBN 0-13-856808-1

Tennant, C, Boonkrong, M. and Roberts, P.A.B., (2002), *The design of a training programme measurement model*, Journal of European Industrial Training Vol. 26/5 pp 230-240 MCB University Press, ISSN 0309-0590

Turner, E. (2006), *Learning that lasts, Industrial and commercial training*, Vol. 38 No. 3, pp. 137-142, Emerald Group Publishing Limited, ISSN 0019-7858.

Van der Wiele T., Dale, B. Carter, G., Kolb F., Luzon D. M., Schmidt A., and Wallace M, (1995), *Self-assessment: A Study of Progress in Europe's Leading Organisations in Quality Management Practice*, International Journal of Quality and Reliability Management, Vol. 13 No 1 pp 84-104 , MCB University Press, ISSN 0265-671X

Vinten, G. (1994), *The Behavioural Aspects of Quality Assurance, Lessons from Internal Auditing,* Quality Assurance in Education, Vol. 2 No. 1, pp. 4-12, MCB University Press, ISSN 0968-4883.

Vinten, G. (1998), *The competition for quality in the audit society*, Managing Service Quality, Vol. 8 No. 6, pp. 389-394, MCB University Press, ISSN 0960-4529.

Vreede G.J. de., (1995), *Facilitating Organisational Change: The Participative Application of Dynamic Modelling.*, Doctoral Dissertation, Delft University of Technology

Waina, R.B. Ph.D. (2001), *Five Critical Questions in Process Improvement*, web site http//www.chips.navy.mil/archives/01_summer/five_critical_questions_in_pro ce..htmvisited April 2006.

Wagner H., (1999), *The Psychobiology of Human Motivation*, Routledge, Taylor & Francis (UK), Abingdon, Oxon., England, ISBN 0-415192-95-7

Wealleans D. (2000), *The Quality Audit for ISO 9001:2000, A Practical Guide,*

Gower Publishing Company, ISBN 0 566 08245 4.

Weinberg G., (1993), *Quality software management, Volume 1 – System Thinking,* Dorset House Publishing, New York, USA, ISBN: 0-932633-24-2

Yin, R.K. (1989) *Case study research: Design and Methods*, Sage Publications, Beverly Hills, California

Yourdon, E., (1989), *Structured Walkthroughs*, Prentice Hall, New Jersey, USA. ISBN 0138552893

Annex I

This checklist was used for the software quality assurance baseline assessment discussed in Chapter 6.

Software Quality Implementation Assessment Checklist

Software Item or Project		Point of Contact	
Area/Project		Date	

Summary Result

Topic	Score	Comment
Assignment of responsibility		
Software Inventory		
Categorisation		
Management Plan		
Specification		
Design, development & Implementation		
Verification/ Reviews		
Testing		
Configuration Management		
Disaster Recovery		
Computer Services		
Procurement		

Organisation

Name of Software Representative	

	Yes, no or n/a
Does a letter of appointment exist?	
Are the responsibilities documented in job description?	
Is the appointment listed in Nominated Representatives Folder?	
Has a review been conducted for the need for local procedures?	
Are local procedures in place?	
Is there evidence available of the review or audit of local procedures?	

Register/inventory

Register Reference	

	Yes / no
Is the register listed in the Company list?	
Has the register up to date?	
Does the register contain all the required entries for each software item?	

Sample Software Project Register check:

Title	Version	Category	Status	Custodian	Proj plan ref

Categorisation

	Yes / no
Is the software categorisation system understood?	
Is there evidence of review of the software categories?	

Assessment of correct interpretation of categorisation system

Software Item	Correct Category Yes / no

Software Control Plan

Software Control Plan Reference Number	

	Yes / no
Is the plan up to date (reviewed within the last three years)?	
Does the plan contain all the required topic heading?	
Are the approval arrangements adequate?	

Template Quality Check

Topic Heading	**Has the topic been adequately covered? Yes / No or n/a**
Line management assessment	
Scope	
Organisation & Responsibilities	
Security	
Specification	
Programme Management	
Design & Development	
Use	
Maintenance	
Review & Audit	
Configuration management	
Problem reporting & corrective action	
Procurement	
Computing environment	
Verification & validation	
Installation	
Records	
Definitions	

Requirements specification

	Yes / no
Does a specification exist?	
Is the specification up to date? (changes have been incorporated)	
Are the approval arrangements adequate?	

Requirements Template Quality Check

Topic Heading	Has the topic been adequately covered? Yes / No or n/a
Introduction and scope	
Business justification	
Requirements Functional	
Requirements non-functional (performance)	
Availability and reliability	
Usability	
Maintainability	
Portability	
System interfaces	
System Constraints	
Installation and commissioning	
Considerations for the future	
Project requirements	
Post delivery and support	

Design, Development & Implementation

	Yes / no or n/a
Is a development life cycle defined?	
Has the right life cycle been chosen?	
Has the need for software design considered?	
Is the design traceable from the requirements?	
Is a coding standard referenced for development?	
Is there evidence of the coding standard being applied?	
Is there evidence of verification of each phase output	
host & target environment	
Have software tools been validated	

Review and Verification

Have the following been reviewed?	Yes / No or n/a	Adequate
Register		
Category(ies) of software item(s)		
Software Control Plan		
Development Contract		
User Requirements Specification		
Software Requirement Specification		
Interface design		
Architectural design		
Detail design		
Data design		
Code against coding standard		
Test Plan		
Test Results		
Configuration management arrangements		

Computer Services

Is there a service plan reference in place?		
Is it up to date?	YES/NO	

Are the following in the service plan?	Yes / No or n/a	Adequate
purpose, service & communication		
organisation & responsibilities		
method for configuration control		
principles for purchases		
installation validation		
fault logging & corrective action		
media control procedures		
registration & roles for users		
service quality (e.g. availability, help & repair)		
Disaster recovery procedure in place		

Procurement

	Yes / No or n/a
Is the software registered?	
Is the software categorised correctly?	
Is there a software control plan in place?	
Are the requirements documented?	
Are the requirements up to date?	
Has evidence of Y2K compliance been requested?	
Is the evidence available?	
Have the deliverables been clearly identified?	
Is the acceptance criteria defined?	
Has post delivery been considered and in place?	
Has the supplier stability for product life been considered?	
Have training requirement been considered?	
Is user documentation needed and available?	
Are test records available?	
Has software been virus checked?	

Annex II

Software Quality Management System Assessment Results 1999

Assessment data collated in 1999 on the implementation of the software quality system in each organisational functional group at AWE plc. The background and analysis of this data is discussed in Section 6.5 of Chapter 6.

Table AII.1. Assessment results of nine software systems in assurance

System	org	reg	cat	scp	spec	des	r&v	t&v	scm	d.r.	c.s.	proc	Total	Max	%imp
SQS1	2	1	3	2	1	1	1	1	1	1	1	1	16	48	33.3%
SQS2	2	2	1	2	1	1	1	2	1	0	n/a	2	15	44	34.1%
SQS3	2	2	1	1	1	1	1	2	0	0	n/a	2	13	44	29.5%
SQS4	2	1	1	1	1	1	1	1	1	3	n/a	1	14	44	31.8%
SQS5	2	2	1	2	1	1	1	2	0	1	n/a	1	14	44	31.8%
SQS6	2	2	1	2	1	1	1	1	0	1	n/a	1	13	44	29.5%
SQS7	0	1	1	1	0	n/a	1	1	1	1	n/a	n/a	7	36	19.4%
SQS8	1	1	1	1	1	1	1	1	0	1	n/a	1	10	44	22.7%
SQS9	2	2	2	1	0	0	0	1	1	1	n/a	n/a	10	40	25.0%
Totals	15	14	12	13	7	7	8	12	5	9	1	9	112	388	
Average	1.67	1.56	1.3	1.44	0.78	0.88	0.9	1.33	0.56	1	1	1.29			28.9%

Table AII.2. Assessment results of two software systems in commercial

System	org	reg	cat	scp	spec	des	r&v	t&v	scm	d.r.	c.s.	proc	Total	Max	%imp
C1	1	1	3	1	1	n/a	1	1	0	2	n/a	3	14	40	35.0%
C2	1	1	1	1	0	0	0	1	1	2	n/a	n/a	8	40	20.0%
Totals	2	2	4	2	1	0	1	2	1	4		3	22	80	
Average	1	1	2	1	0.5	0	0.5	1	0.5	2		3			27.5%

Table AII.3. Assessment results of nine software systems in infrastructure

System	org	reg	cat	scp	spec	des	r&v	t&v	scm	d.r.	c.s.	proc	Total	Max	%imp
INF1	3	2	3	2	1	1	1	2	1	1	n/a	n/a	17	40	42.5%
INF2	3	2	3	3	2	1	1	2	2	2	n/a	2	23	44	52.3%
INF3	3	2	3	3	2	1	1	2	2	2	n/a	n/a	21	40	52.5%

INF4	2	2	3	2	1	1	1	2	0	2	n/a	n/a	16	40	40.0%
INF5	2	2	4	3	2	1	1	2	1	2	n/a	1	21	44	47.7%
INF6	2	2	4	3	2	1	1	2	1	2	n/a	1	21	44	47.7%
INF7	2	1	2	2	1	1	1	1	1	2	n/a	1	15	44	34.1%
INF8	2	2	3	2	1	1	0	1	1	2	n/a	n/a	15	40	37.5%
INF9	2	2	3	2	1	1	0	1	1	2	n/a	n/a	15	40	37.5%
Totals	21	17	28	22	13	9	7	15	10	17		5	164	376	
Average	2.33	1.89	3.11	2.44	1.44	1	0.8	1.67	1.11	1.89		1.25			43.6%

Table AII.4. Assessment results of four software systems in finance

System	org	reg	cat	scp	spec	des	r&v	t&v	scm	d.r.	c.s.	proc	Total	Max	%imp
F1	1	1	1	1	1	0	1	2	1	1	1	2	13	48	27.1%
F2	1	1	1	1	1	n/a	0	2	0	1	n/a	1	9	40	22.5%
F3	1	1	1	1	1	1	0	3	1	2	n/a	1	13	44	29.5%
F4	1	1	1	1	1	n/a	1	2	0	2	n/a	1	11	40	27.5%
Totals	4	4	.4	4	4	1	2	9	2	6	1	5	46	172	
Average	1	1	1	1	1	0.5	0.5	2.25	0.5	1.5	1	1.25			26.7%

Table AII.5. Assessment results of two software systems in personnel

System	org	reg	cat	scp	spec	des	r&v	t&v	scm	d.r.	c.s.	proc	Total	Max	%imp
P1	3	1	1	1	0	n/a	0	1	0	2	n/a	1	10	40	25.0%
P2	3	2	1	1	1	0	0	1	1	1	n/a	0	11	44	25.0%
Totals	6	3	2	2	1	0	0	2	1	3		1	21	84	
Average	3	1.5	1	1	0.5	0	0	1	0.5	1.5		0.5			25%

Table AII.6. Assessment results of eleven software systems in technology

System	org	reg	cat	scp	spec	des	r&v	t&v	scm	d.r.	c.s.	proc	Total	Max	%imp
T1	3	2	3	2	1	n/a	1	2	1	2	n/a	2	19	40	47.5%
T2	3	2	3	2	1	0	0	2	1	2	n/a	n/a	16	40	40.0%
T3	2	1	2	2	1	0	0	1	1	0	n/a	1	11	44	25.0%
T4	3	1	3	3	2	2	1	1	1	2	n/a	n/a	19	40	47.5%
T5	3	1	2	2	1	1	1	1	0	2	n/a	1	15	44	34.1%
T6	3	1	3	2	0	0	0	1	1	2	n/a	n/a	13	40	32.5%
T7	3	1	2	2	0	0	0	2	1	2	n/a	n/a	13	40	32.5%

T8	2	2	3	3	1	n/a	0	2	0	2	n/a	2	17	40	42.5%
T9	2	1	2	2	1	n/a	1	1	0	1	n/a	n/a	11	36	30.6%
T10	3	1	3	3	1	1	1	2	1	1	1	1	19	48	39.6%
T11	4	4	4	2	1	n/a	1	2	1	2	n/a	1	22	40	55.0%
Totals	31	17	30	25	10	4	6	17	8	18	1	8	175	452	
average	2.82	1.55	2.72	2.27	0.91	0.57	0.5	1.55	0.73	1.64	1	1.33			38.7%

Table AII.7. Assessment results of nine software systems in weapons

System	org	reg	cat	scp	spec	des	r&v	t&v	scm	d.r.	c.s.	proc	Total	Max	%imp
W1	2	4	3	4	3	2	3	3	2	2	n/a	n/a	28	40	70.0%
W2	2	2	2	2	2	2	2	3	2	2	n/a	n/a	21	40	52.5%
W3	0	0	0	0	0	0	0	0	0	0	n/a	0	0	44	0.0%
W4	3	4	4	3	3	3	3	3	2	0	n/a	n/a	28	40	70.0%
W5	3	3	4	3	3	3	2	3	2	0	n/a	n/a	26	40	65.0%
W6	2	2	4	3	2	0	2	2	2	0	n/a	2	21	44	47.7%
W7	2	2	4	3	2	0	2	2	1	0	n/a	1	19	44	43.2%
W8	2	1	2	2	1	0	2	2	1	1	n/a	1	15	44	34.1%
W9	4	3	2	4	4	n/a	2	4	1	2	n/a	n/a	26	36	72.2%
Totals	20	21	25	24	20	10	18	22	13	7		4	184	372	
Average	2.22	2.33	2.77	2.67	2.22	1.25	2	2.44	1.44	0.78		1			49.5%

Table AII.8. Assessment results of nine software systems in research

System	org	reg	cat	scp	spec	des	r&v	t&v	scm	d.r.	c.s.	proc	Total	Max	%imp
R1	3	2	2	3	2	1	1	2	1	1	n/a	n/a	18	40	45.0%
R2	2	2	2	1	0	0	0	1	1	0	n/a	n/a	9	40	22.5%
R3	2	2	2	2	1	1	1	3	2	2	n/a	n/a	18	40	45.0%
R4	1	1	2	1	1	0	0	2	0	1	n/a	1	10	44	22.7%
R5	2	1	1	1	0	n/a	0	1	1	1	n/a	n/a	8	36	22.2%
R6	2	2	2	1	1	1	1	1	1	2	n/a	n/a	14	40	35.0%
R7	2	2	2	1	1	1	1	2	1	1	n/a	1	15	44	34.1%
R8	3	1	1	2	2	1	1	1	1	1	1	n/a	15	44	34.1%
R9	3	2	3	3	2	0	0	1	0	1	n/a	n/a	15	40	37.5%
Totals	20	15	17	15	10	5	5	14	8	10	1	2	122	368	
Average	2.22	1.67	1.88	1.67	1.11	0.63	0.6	1.56	0.89	1.11	1	1			33.1%

Annex III

Self-Assessment Spreadsheet

This spreadsheet listing was used for facilitated self-assessment discussed in Chapter 7. It could also be used by delegates as part of their training conveyed in Chapter 8 and for the purposes of monitoring the overall Company improvements in implementation of the software quality management system highlighted in Chapter 10.

Software Management & Control Plans, Measures and Staff Effectiveness

Compliant	Applicable Y=1/N=0	Met Y=1/N=0
Is the software registered?	0	0
Does the register contain all minimum required entries?	0	0
Are all entries complete and up to-date?	0	0
Is the complete baseline listed or referenced?	0	0
Has the directorate assurance group been notified of the register & details?	0	0
Have risks been assessed and the software correctly categorised?	0	0
Does a Software Management & Control Plan exist?	0	0
Is the Software Management and Control Plan up to date?	0	0
Are all required sections of the SMCP been addressed?	0	0
Has the SMCP been reviewed before issue?	0	0
Is there evidence of review?	0	0
Totals	0	0
Compliant and of appropriate quality % Implemented =	0	%

- Is there evidence that.....	**Applicable** Y=1/N=0	**Met** Y=1/N=0
the SMCP is regularly reviewed? (Minimum 3 years)	0	0
the SMCP has up dated when significant changes have taken place?	0	0
Totals	0	0
Established and adding value % Implemented =	0	%

Total score = (75% (Compliant/Quality) + 25%(Established/Value)) =	0	%

Exceptional Practices

Staff Effectiveness	**Applicable** Y=1/N=0	**Met** Y=1/N=0
Has the competency framework been considered for all staff involved in software?	0	0
Have all the considered competencies been met for all staff?	0	0
Total	0	0
% Implemented	0.00%	

Process Measures	Applicable Y=1/N=0	Met Y=1/N=0
Have process and product measures been considered?	0	0
Are measurement systems in place to enable improvements?	0	0
Have measures been used to improve product quality or process improvement?	0	0
Is information captured on the effectiveness of the review process?	0	0
Total	0	0
% Implemented	0.00%	

Requirements Specification

Compliant	Applicable Y=1/N=0	Met Y=1/N=0
Does a requirements specification or list of user requirements exist?	0	0
Does the specification include functional and performance requirements?	0	0
Has the requirements specification been reviewed	0	0
Is there evidence of that review	0	0
Is the specification traceable through to testing?	0	0
Totals	0	0

Compliant % Implemented = 0 **%**

Established	Applicable Y=1/N=0	Met Y=1/N=0
- Is there evidence that.....		
the specification has been updated after a change request ?	0	0
the requirements remain traceable to testing after changes?	0	0
Totals	0	0

Established % Implemented = 0 **%**

Total score = (75% (Compliant) + 25%(Established)) = 0 **%**

Exceptional Practices

	Applicable Y=1/N=0	Met Y=1/N=0
Have requirements been elicited from all stakeholders?	0	0
Has the specification been reviewed & agreed by all stakeholders?	0	0
Has a requirements traceability matrix been developed?	0	0
Is a requirements capture method documented & followed?	0	0
Has requirements capture method has been maintained for changes?	0	0
Total	0	0
% Implemented	0.00%	

Configuration Management

Compliant	Applicable Y=1/N=0	Met Y=1/N=0
Is there a documented system for how the local area manages SCM?	0	0
Are all configuration management responsibilities defined?	0	0
Are all configured items uniquely identified?	0	0
Is the status of each item documented?	0	0
Is there a list (baseline) of items that describes what the system consists of?	0	0
Is there a system that captures change requests & problems encountered?	0	0
Are change requests reviewed for impact on the baseline?	0	0
Does the problem reporting system identify root cause?	0	0
Are measures in place to review severity and frequency of problems?	0	0
If changes affect facilities is the 804 system used?	0	0
Is a test spec produced for changes with pass/fail criterion?	0	0
Is regression (repeated system) testing conducted at appropriate times?	0	0
Has the development history of each configured item been recorded?	0	0
Has the documented configuration been checked the actual system in use?	0	0
Totals	0	0

Compliant % Implemented = 0.00 **%**

Established - Is there evidence that.....	Applicable Y=1/N=0	Met Y=1/N=0
baselines are up dated as changes occur?	0	0
integrity checks (audits) are regularly conducted?	0	0
Totals	0	0

Established % Implemented = 0.00 **%**

Total score = (75% (Compliant) + 25%(Established)) = 0.00 **%**

Exceptional Practices

	Applicable Y=1/N=0	Met Y=1/N=0
Correcting the root cause of problems leads to process improvement?	0	0
the frequency of problems are tracked, assessed and controlled (e.g. reducing) ?	0	0
Has the configuration management system been automated in a CM Tool?	0	0
Total	0	0
% Implemented	0.00	

Development - Design & Coding Practice

Compliant	Applicable Y=1/N=0	Met Y=1/N=0
Is the development life cycle or process defined?	0	0
Has each phase been reviewed and verified?	0	0
Are requirements developed into:		
Interfacing, architectural	0	0
detail and data design (as required by evaluation of quality characteristics)?	0	0
Has design been reviewed for feasibility, consistency, and traceability?	0	0
There is evidence of review?	0	0
Has a coding standard been defined for the language used?	0	0
Has the source code been reviewed against the coding standard?	0	0
Is there procedure for compiling and build?	0	0
Are arrangements for backing up development code defined and met?	0	0
Totals	0	0
Compliant % Implemented =	**0**	**%**

Established **- Is there evidence that.....**	Applicable Y=1/N=0	Met Y=1/N=0
design information has been updated for changes ?	0	0
the coding standard is adhered to for change implementations	0	0
Totals	0	0
Established % Implemented =	**0**	**%**

Total score = (75% (Compliant) + 25%(Established)) =	**0**	**%**

Exceptional Practices

	Applicable Y=1/N=0	Met Y=1/N=0
Have quality characteristics been evaluated & decisions captured on design?	0	0
Has a design to requirements traceability matrix been developed?	0	0
Has a design methodology been utilised?	0	0
the design methodology has been maintained for changes (after acceptance)	0	0
Are inspection techniques used to capture process efficiency for design?	0	0
Are CASE Tools used to automate the design process?	0	0
Has the code been developed by people trained in good programming techniques?	0	0
Has the code been developed by people trained in the applicable language?	0	0
Has a code walkthrough been conducted?	0	0
Can the requirements be identified within the code headers?	0	0
Is traceability within the code, from requirements, design through to testing?	0	0
Has the software been developed with reuse objectives?	0	0
Total	0	0
% implemented	0.00%	

Testing, Validation and Acceptance - SW1907

Compliant	Applicable Y=1/N=0	Met Y=1/N=0
Is a test plan in place?	0	0
Are the stages, sequencing and responsibilities of testing clear?	0	0
Is the description of the testing environment sufficient for independent replication?	0	0
Is there a test for each requirement?	0	0
Does each test carry pass or fail criteria?	0	0
Are test results recorded, analysed and available?	0	0
Is system (regression) testing defined and performed at appropriate times?	0	0
Is software reliability considered by the testing of range or boundary conditions?	0	0
Do test records provide traceability for versions, input data and authorising action?	0	0
Are limitations for use described?	0	0
Totals	0	0

Compliant % Implemented = 0 **%**

Established - Is there evidence that.....	Applicable Y=1/N=0	Met Y=1/N=0
tests plans, procedures, specs and results are updated when changes occur ?	0	0
use with limitations issues resolved ?	0	0
Totals	0	0

Established % Implemented = 0 **%**

Total score = (75% (Compliant) + 25%(Established)) = 0 **%**

Exceptional Practices

	Applicable Y=1/N=0	Met Y=1/N=0
Can the statement, path and branch coverage be ascertained?	0	0
Are tools used to automate the testing process	0	0
Total	0	0

% Implemented 0.00%

Operational Use - SW1902 Section 4.7

Compliant	Applicable Y=1/N=0	Met Y=1/N=0
Are procedures, work instructions or manuals for using software produced?	0	0
Are user organisations, roles & responsibilities defined?	0	0
Have competency considerations been met?	0	0
Access control arrangements defined?	0	0
Is any user related back-up arrangements defined?	0	0
Is there a system to raise changes and problems?	0	0
Are affected personnel notified of a change by an appropriate method?	0	0
Does problem reporting assess: impact, severity, frequency & root cause?	0	0
Totals	0	0

Compliant % Implemented = 0 **%**

Established - Is there evidence that.....	Applicable Y=1/N=0	Met Y=1/N=0
procedures, instructions or guides are updated when changes occur?	0	0
user lists and responsibilities have been updated when roles change ?	0	0
that back-ups occur regularly and checked ?	0	0
Totals	0	0

Established % Implemented = 0 **%**

Total score = (75% (Compliant) + 25%(Established)) = 0 **%**

Exceptional Practices

	Applicable Y=1/N=0	Met Y=1/N=0
Is the analysis of problems leading to process improvements?	0	0
Total	0	0

% Implemented 0.00%

Computer System Disaster Recovery Plan SW1908

Disaster Protection and Disaster Recovery

Compliant	Applicable Y=1/N=0	Met Y=1/N=0
Have assessment of risks been conducted?	0	0
Is a Disaster Recovery Plan in place?	0	0
Does the contents of disaster recovery plan include:		
Responsibilities defined	0	0
risks identified	0	0
Impact of disaster	0	0
Operational requirements	0	0
protective measures		
backup	0	0
access control	0	0
physical protection	0	0
Contingency arrangements	0	0
disaster recovery arrangements	0	0
Has the plan been tested or is there written justification for not testing?	0	0
Totals	0	0

Compliant % Implemented = 0 %

Established - Is there evidence that.....	Applicable Y=1/N=0	Met Y=1/N=0
the disaster recovery plan has been reviewed & kept up to date?	0	0
back-ups occur regularly and are checked?	0	0
tests have been conducted after a change to the system?	0	0
Totals	0	0

Established % Implemented = 0 %

Total score = (75% (Compliant) + 25%(Established)) = 0 %

Computer Services - SW1906

Compliant	Applicable Y=1/N=0	Met Y=1/N=0
Is there a service plan in place?	0	0
Does the contents of Service Plan include :		
Purpose, service & communication	0	0
Organisation & responsibilities	0	0
Method for configuration control	0	0
Purchasing principles	0	0
Method for installing, validating & notifying users of change	0	0
Media control procedures (inc. backup)	0	0
Methods for fault logging & corrective action	0	0
Registration of users & control of access privileges	0	0
Service quality (e.g. availability, helpdesk & repair time)	0	0
Is a baseline or system description in place for the current installation?	0	0
Are proposed changes analysed for impact, reviewed and agreed?	0	0
Are users notified of planned outages or updates?	0	0
Is service provision performance monitored?	0	0
Is a disaster recovery procedure in place?	0	0
Totals	0	0

Compliant % Implemented = 0 %

Established - Is there evidence that.....	Applicable Y=1/N=0	Met Y=1/N=0
the plan has been reviewed and updated when changes occur ?	0	0
service provision performance has been reviewed and any action taken?	0	0
the system baseline has been updated regularly ?	0	0
Totals	0	0

Established % Implemented = 0 %

Total score = (75% (Compliant) + 25%(Established)) = 0 %

Procurement - SW1905

Compliant	Applicable Y=1/N=0	Met Y=1/N=0
Is a Software Management & Control Plan in place?	0	0
Does the order include specified requirements?	0	0
Have quality conditions been stated?	0	0
Have all deliverables been specified? (e.g. media, documentation)	0	0
Has testing, validation and acceptance arrangements been defined?	0	0
Have the following been considered:		
Installation/commissioning	0	0
Supplier support (post delivery)	0	0
Supplier's commercial stability	0	0
Procedure for releasing to users	0	0
Maintenance and change control arrangements	0	0
Training needs	0	0
Escrow arrangements	0	0
Have any special arrangements with the supplier been defined & agreed?	0	0
Has the software been validated and accepted before use?	0	0
Has the software been virus checked before installation and use?	0	0
Software Requiring development		
Has a Joint customer/supplier review taken place?	0	0
Is a Quality plan (similar to SMCP) been provided by supplier?	0	0
Has a statement been made that IPR are vested in MOD?	0	0
Are external suppliers TickIT certified or has justification given?	0	0
Totals	0	0

Compliant % Implemented = 0 %

Established - Is there evidence that.....	Applicable Y=1/N=0	Met Y=1/N=0
the SMCP has been up dated for system changes?	0	0
the system specification remains current?	0	0
testing arrangements and records are repeatable and available?	0	0
system baseline have been updated for changes?	0	0
Totals	0	0

Established % Implemented = 0 %

Total score = (75% (Compliant) + 25%(Established)) = 0 %

Software System	Issue	Assessment Ref.:

Topic	Applicable Y=1/N=0	Score	Supporting Comments
Basic Control	1	0	
Requirements Specification	1	0	
Development	1	0	
Testing, validation & acceptance	1	0	
Use	1	0	
Configuration Management	1	0	
Disaster Recovery	1	0	
Computer Services	1	0	
Procurement	1	0	
Totals	**9**	**0**	

Final Result = $\frac{\text{Total score}}{\text{Total Applicable}}$ **0** %

Software System	Issue	Assessment Ref.:

Topic	Applicable Y=1/N=0	Score	Supporting Comments
Basic Control	1	0	
Requirements Specification	1	0	
Development	1	0	
Testing, validation & acceptance	1	0	
Use	1	0	
Configuration Management	1	0	
Disaster Recovery	1	0	
Computer Services	1	0	
Procurement	1	0	
Totals	**9**	**0**	

Final Result = $\frac{\text{Total score}}{\text{Total Applicable}}$ **0** %

Annex IV

Software Quality Management System Assessment Results 2006 and 2007

Tables of software system assessments undertaken in 2006 and 2007 at AWE plc and analysed in Chapter 10. Results in bold are below 50% and discussed in Chapter 10.

Table A IV.1 Audit Assessment Results for 2006

System Number	Initial Assessment	Improvement %	Final Percentage Implemented
1	57%	21%	78%
2	57%	21%	78%
3	61%	0%	61%
4	**34%**	53%	87%
5	52%	0%	52%
6	52%	0%	52%
7	68%	10%	78%
8	68%	10%	78%
9	64%	12%	76%
10	57%	21%	78%
11	57%	21%	78%
12	57%	21%	78%
13	57%	21%	78%
14	**32%**	64%	96%
15	**47%**	47%	94%
16	68%	10%	78%
17	52%	0%	52%
18	52%	0%	52%
19	68%	10%	78%
20	57%	21%	78%
21	57%	21%	78%
22	68%	10%	78%
23	68%	10%	78%
24	**46%**	12%	58%
25	**46%**	12%	58%
26	88%	0%	88%
27	70%	0%	70%
28	74%	0%	74%
29	54%	0%	54%
		Company Average	**73%**

Table A·IV.2 Audit Assessment Results for 2007

System Number	Initial Assessment	Improvement %	Final Percentage Implemented
1	78%	14%	92%
2	78%	14%	92%
3	63%	13%	76%
4	78%	14%	92%
5	78%	14%	92%
6	59%	23%	82%
7	78%	14%	92%
8	59%	29%	88%
9	78%	14%	92%
10	78%	14%	92%
11	52%	37%	89%
12	59%	20%	79%
13	52%	37%	89%
14	78%	14%	92%
15	59%	20%	79%
16	59%	27%	86%
17	59%	21%	80%
18	78%	14%	92%
19	78%	14%	92%
20	78%	14%	92%
21	78%	14%	92%
22	46%	12%	58%
23	46%	12%	58%
24	88%	0%	88%
25	73%	5%	78%
26	59%	25%	84%
27	59%	27%	86%
28	59%	29%	88%
29	38%	50%	88%
30	38%	50%	88%
31	74%	0%	74%
32	59%	29%	88%
33	59%	21%	80%
34	59%	25%	84%
35	59%	29%	88%
36	59%	27%	86%
37	59%	27%	86%
38	59%	21%	80%

System Number	Initial Assessment	Improvement %	Final Percentage Implemented
39	52%	37%	89%
40	78%	14%	92%
41	59%	37%	96%
42	59%	29%	88%
43	59%	11%	70%
44	52%	37%	89%
45	78%	14%	92%
46	59%	25%	84%
47	59%	21%	80%
48	59%	25%	84%
49	59%	25%	84%
50	59%	25%	84%
51	54%	20%	74%
		Company Average	**81.0%**