



This item was submitted to Loughborough's Institutional Repository by the author and is made available under the following Creative Commons Licence conditions.



CC creative commons
COMMONS DEED

Attribution-NonCommercial-NoDerivs 2.5

You are free:

- to copy, distribute, display, and perform the work

Under the following conditions:

BY: **Attribution.** You must attribute the work in the manner specified by the author or licensor.

Noncommercial. You may not use this work for commercial purposes.

No Derivative Works. You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the [Legal Code \(the full license\)](#).

[Disclaimer](#) 

For the full text of this licence, please go to:
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

An Analysis of Software Quality Management at AWE plc.

Michael Elliott¹, Ray Dawson², Janet Edwards²

¹Software Quality Manager, AWE plc., Aldermaston, Reading, Berkshire. UK

²Department of Computer Science, Loughborough University, Loughborough, UK

Abstract

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 paper presents the question set used and the subsequent results obtained from the implementation assessment for 55 software systems at AWE plc. The data is 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.

1. Introduction

The two principal aims of this research paper 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 one of the authors¹ 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. The data presented in this paper has 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 in understanding the issues and problems associated with implementation.

2. Background

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. In 1993, AWE became a government owned, contractor operated establishment and since then has been managed by two large contractor consortiums. The assessments for this research were conducted during the first period of contractor management. The site management and operations are still monitored at different levels by project teams from the MoD.

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. 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 (res) with highly qualified scientific personnel, a weapons or Systems Engineering group (weap) with the major concentration of highly qualified engineers, a production technology group (tech) which has a mix of qualified engineers and scientific staff, and the a site development group (inf) providing the site infrastructure and support with less qualified, but highly skilled engineering

workforce. Support activities are also provided by a smaller number of staff in support groups, namely, personnel (pers), financial (fin), and commercial (com). Safety, quality, and the security functions were amalgamated in an assurance group (sqs).

A key organisational issue that had 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 assignment of a risk category to each software item or product. The software category defines the control activities required for its subsequent management. There was no consistency to the type of people appointed as LASRs, which varied from administrative, scientific, engineer, developer or staff with other quality system roles. This would have been 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 to other groups as well as the IT network infrastructure, but 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.

3. Literature Review

This 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.

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 (CMM) (Carnegie, 2000).

The guidance (ISO, 2000) to the revision of the ISO 9001 standard from 1994 to 2000 highlighted 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 CMM 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.

Organisational culture does not feature highly in these standards or reports, although competency normally encompasses motivation (Spencer et al, 1992) along with knowledge and experience. Motivation can be influenced by cultural factors (Wagner, 1999). Indeed many excellent technical books on software quality management do not focus on culture, only occasionally referencing the issue. Horch (1996) warns of the pitfalls, "starting a software quality programme is doomed to failure – inadequate preparation, misused terms, lack of planning, failure to recognise the individual role of members of the organisation are only a few pitfalls". 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 to them. However, this is known to be not totally successful and a further review to establish the need for local procedures may be necessary to gain full commitment to implementation. 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 underestimated (Siaksa, 2004). This situation may be further complicated in a multi-national company (Siaksa, 2004). Indeed, the style of management (Parzinger, 2001) also has a significant impact on culture

Despite this lack of emphasis on cultural issues for quality management adoption and recognition of its impact has been researched and made available. Crosby (1997) was an early pioneer who noticed coherent patterns of culture impacting on technology processes. Radice et al (1984) 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 1. However, Weinberg warns that maturity may not be the right word, as any software culture can be successful. Indeed, moving from one cultural pattern, as Weinberg prefers, may not be right for certain organisations.

Table 1. 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

4. 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 a 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; (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 assessment data was then collated by topic area and by main functional group to see if there were specific improvement opportunities or cultural issues that needed addressing.

5. 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 2. 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 the data was gathered as part of the Company audit programme. 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.

Table 2. 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

The subject areas in which the company software quality management system was divided are listed in Table 3. 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 3. 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. Results

Tables 4 to 11 are the 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 2) was assigned as that value for each software quality topic (Table 3). 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 12 together with the overall Company average. Table 13 presents the overall Company average for each software quality topic to indicate the strengths and weaknesses of implementation areas. The tasks that are the least undertaken for each software quality topic are contained in Table 14. The Local Area Software Representatives (LASRs) survey on how well they knew their roles and how they felt about it are conveyed in Table 15. Table 16 is a list of the types of comments received when conducting the assessments on the implementation of the software quality management system.

Table 4. Assessment results of nine software systems in assurance (sqs)

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 5. Assessment results of two software systems in commercial (com)

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 6. Assessment results of nine software systems in infrastructure (inf)

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 7. Assessment results of four software systems in finance (fin)

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 8. Assessment results of two software systems in personnel (pers)

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 9. Assessment results of eleven software systems in technology (tech)

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 10. Assessment results of nine software systems in weapons (weap)

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 11. Assessment results of nine software systems in scientific research (res)

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%

Table 12. Average assessment results for each of the functional groups

Group	sqs	com	inf	fin	pers	tech	weap	res	Company
Average % implementing rating	28.7	27.5	43.62	26.74	25	38.72	49.46	33.15	34.1%

Table 13. Average assessment results for each topic across the Company

Topic	org	reg	cat	scp	spec	des	r&v	t&v	scm	d.r.	c.s.	proc
Ave Co	2.03	1.56	1.98	1.69	1.06	0.6	0.72	1.6	0.78	1.43	1	1.33
Imp rate												

Table 14. 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 accepted
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 15. 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 16. 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,	Often	Occasionally	Rarely

can't you provide a diagram?			
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

7. Analysis of results

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.

The assurance group (sqs) results from Table 4 show that the organisational arrangements attracted the highest scores on the implementation rating. Within this topic, the actions of appointing of local areas software representatives (LASRs) with appropriate job descriptions were generally undertaken well. All LASRs 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 (com) only owned two software systems, as documented in Table 5, 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 6 shows the infrastructure group (inf) achieved higher scores on the assignment of the correct category for software systems. 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. The production of control plans scored well.

The financial group (fin) scored highest on testing and validation (t&v) as can be seen from Table 7. This is in keeping with this groups affinity to get the numbers right. A lack of technical knowledge resulted in low levels being recorded on most other subjects.

Predictably the personnel group (pers) scores in Table 8 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 their low scores.

The eleven systems assessed in the technology area (tech) provided the biggest variance of results. Total absence of specification and review records, together with a lack of design were the main problems areas, as can be seen in Table 9. The appointment of LASRs had been completed well and the production of control plans was 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 (weap) as shown in Table 10. 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 of their systems. However, this was an organisational group where the maintenance of the quality system was known to be good and seen to add value as emphasised by the number of fours recorded.

The research group (res) has some very large and complex software and the need for design information was a hot topic of debate, which Table 11 shows had a low score. 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.

As can be seen from Table 12, 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, it was clearly 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.

A list of the strongest and weakest topics, as assessed and averaged out across the whole Company, can be seen in Table 13. The weakest implemented topics, together with the list of least undertaken activities in Table 14, provides a succinct training needs analysis, highlighting the topics requiring significant education and supporting guidance documentation.

The group comments recorded in Table 16 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), which 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 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 in Table 16, because of the slightly different type of comments they 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 science and 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 it has not been documented. 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 group demonstrated a high degree of willingness to achieve compliance but not always a fully understanding 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.

8. 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.

9. 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 (Weinberg, 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 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 Table 13 and the least undertaken tasks in Table 14. 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 15. 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 all support groups (sqs, fin, com, pers) 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.

10. 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. It may 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. 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.

The implementation rating measurement system has demonstrated that it is an excellent 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 paper has shown that the cultural aspects need to be understood in developing an effective strategy for improvement.

REFERENCES

- Carnegie Mellon University, (2001), *Capability Maturity Model Integration*, Software Engineering Institute, Pittsburgh USA.
- 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
- Elliott, M., Dawson, R.J. and Edwards, J., (2005), '*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
- Galín D., (2004), *Software quality assurance, from theory to implementation*, Pearson Education Limited, Essex England, ISBN 0-201-70945-7
- 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
- ISO 9001:2000, (2000), ISO 9001 : 2000 *Quality Management Systems – Requirements*, International Standards Organisation, Geneva, Switzerland.

ISO 9004:2000, (2000), ISO 9001 : 2000 *Quality Management System – Guidelines for performance improvements*, International Standards Organisation, Geneva, Switzerland

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

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

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

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.

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)

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

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

Any views expressed are those of the authors, and do not necessarily represent those of AWE/UK MoD