



**QUALITY CERTIFICATION: LESSONS
FROM THREE SOFTWARE DEVELOPMENT
ORGANISATIONS**

Megan Seen & Anne Rouse

*Working Paper 28/98
March 1998*

ABSTRACT

This paper reports on three case studies of software development organisations that gained certification to the quality standard ISO 9001. It distils fourteen key points learned from the studies, and concludes that software quality certification does have the potential for considerable benefits. These include improved efficiency, higher quality software in certain circumstances, improved budgeting and scheduling, improved use of human resources, and, indirectly, greater customer satisfaction. The studies also establish that certification is not without its costs, and must be embedded within a strong management framework if these benefits are to be realised.

The research involved a review of literature from which a survey instrument was developed. This was used in the pilot study, and was then modified to accommodate highlighted issues. Data was principally gathered through semi-structured interviews with the quality manager and a systems analyst and programmer at the organisations. The data was validated by providing participants with summary reports and subsequent papers for review and comment.

KEYWORDS

quality management system; benefits; costs; Price Waterhouse; MITS; Aspect Computing

MAKING SENSE OF QUALITY CERTIFICATION: LESSONS FROM THREE SOFTWARE DEVELOPMENT ORGANISATIONS

INTRODUCTION

Before a software development organisation is engaged, potential purchasers need to be assured of the quality of the software that will be developed. Yet software quality is very difficult for non-information technology specialists to evaluate. Business managers confronted with the need to decide between competing vendors (and internal service providers) look for evidence that the provider organisation can reliably deliver quality software. Such evidence might include previous experience, industry benchmarks, references from other sites, and the way the staff of the software provider interact with client staff. Other evidence might include certification to one of the various quality standards that are currently in operation.

In Australia, a range of quality assurance programmes exist, including vendor-based programs (such as Novell or Microsoft certification), the SEI's Capability Maturity Model (CMM), and registration as a 'government endorsed supplier'. For the Australian software development industry, one of the more frequently sought signals of quality is certification to the international standard ISO 9001.¹ There are varying opinions concerning the value of certification. Some argue that certification increases competitive advantage (Curtis & Paulk 1993; Tan & Yap 1994) and can help maintain industry position (Buckler 1993; Inwood 1994; Zampetakis 1994). Some believe it improves software quality (Curtis & Paulk 1993; Tan & Yap 1994) and improves the use of resources (Lee & Kim 1992; Perry 1992) while cutting down costs (Bush 1993). Others argue that certification is a costly, bureaucratic, time-consuming process that chiefly involves battling red tape (Buckler 1993; Zampetakis 1994). These critics point out that certification merely documents that certain processes have occurred, and that this may or may not be related to the quality of the overall product that is produced (Glass 1991; Hoyle 1994; Ryan 1994). It is also argued that applying standards can cause loss of creativity (Perry 1992; Sanders & Curran 1995; Schulmeyer 1992; von Hellens 1995) and loss of flexibility (Baker and Rouse 1996; von Hellens 1995).

Large amounts of resources in terms of time, money and people have been invested by companies wishing to gain certification (Zampetakis 1994). This indicates that at least some senior managers believe there is much to be gained by investing in certification, or conversely, that *not* investing in certification may put their firm at a competitive disadvantage.

Given the conflicting views, and the scarcity of empirical evidence about the costs and benefits of software certification, the authors began a series of studies in 1994 into the experiences of Australian organisations who had sought software quality certification. Analysis of the individual case studies and different aspects of these case studies have been reported in other papers (Baker, 1996; Baker and Rouse, 1996a; Baker and Rouse, 1996b). This paper, however, presents the overall implications arising from the findings of these case studies. It starts by describing the studies and the methodology used in the research. It then goes into the implications of the research: potential benefits of the certification process; the way in which certification leads to those benefits; the limitations of software quality certification, and critical implementation issues. The conclusion emphasises the conditions under which benefits to the organisation are most likely to be realised.

The Studies

This paper summarises the results of three case studies conducted within organisations that had achieved certification to ISO 9001 and the ISO-based Australian standard AS 3563 *Software Quality Management System*. The purpose of these studies was to document the benefits and costs (or negative outcomes) of certification to the quality standard. The selected organisations all developed software as their core business, although one did so as part of a larger consulting business. A description of ISO 9001, AS 3563 and associated concepts is not included in this paper, as this has been presented elsewhere (Baker & Rouse, 1996a).

The case studies were designed to answer the following questions:

- What benefits can software development organisations expect to receive from quality certification?
- What are the likely costs or negative outcomes of the certification process?

The organisations chosen for the case studies were Price Waterhouse Urwick² (PWU), Aspect Computing Pty Ltd (Aspect), and Managed Information Technology Solutions (MITS). While these organisations operate at many different sites across Australia, information in the case studies relates to the Melbourne office of each company.

Methodology

The approach for the research involved a detailed review of the literature on quality standards, software development and software quality, from which an interview instrument was developed. This instrument was used in the initial pilot study at PWU, which identified further issues, and also highlighted the issues that were perceived to be particularly relevant to practitioners. The instrument was amended to ensure this practitioner focus, and then used within Aspect and MITS.

Semi-structured interviews were carried out with the quality managers, and with selected project managers and systems analysts in each company. Company documents - newsletters, procedures, quality manuals and technical documentation - were also examined. Supporting material was also collected from a former quality manager of one company, from Standards Australia (the largest standards certification body in Australia), from state government purchasing staff, and from support staff within the three organisations. Interviews were taped and subsequently transcribed. With common themes identified, a detailed analysis was carried out searching for data to both support and disconfirm the initial findings. The data was validated by providing respondents with summary reports and subsequent research papers for review and comment. Quantitative data was sought, but not generally available, due to the scarcity of statistical records. (More than one respondent pointed out the difficulty of developing metrics and measuring the degree of improvement in software projects that are unique).

IMPLICATIONS FROM THE CASE STUDIES

A number of common themes emerged from the case studies. These relate to the outcomes of the certification process, the way in which certification leads to those outcomes, the limitations of software quality certification, and critical implementation issues. These themes are described below in the form of conclusions which can be drawn from the experiences of these three companies. More detailed discussion of the evidence on which these conclusions are based can be found in Baker (1996), Baker and Rouse (1996a) and Baker and Rouse (1996b). Because PWU was used as a pilot site to test the methodology and to surface issues, evidence from that site does not appear in all points in the following discussion.

Benefits of Software Quality Certification

Our three case studies show that software quality certification *can* result in a number of positive outcomes: improved software quality, improved efficiencies, and improved customer satisfaction.

Conclusion 1.1: Certification can improve software quality if an organisation has limited experience with a formal development system.

The organisations studied had different levels of commercial software development experience. Those organisations with long-established commercial software development histories (PWU and Aspect) had well-documented development methodologies and procedures. In these organisations, the process of certification appeared to have limited effect on the quality of the final software product; this was seen to be at a high standard already. Gillies (1992: 218) points out that '...benefits [of standards] are likely to be least where good practice already exists'. In contrast, the newer organisation, MITS, described more specific improvements in terms of reliability, testability, maintainability and correctness of the software as a result of the changes it made in response to certification requirements. This points to certification having greater

value to those organisations with less experience in using a systematic development approach. This would seem to bear out the views of Deming (1986), Humphrey (1989) and Curtis and Paulk (1993) who held that the use of standards will improve product quality because standards help to improve the process through which the product is created.

Conclusion 1.2: Certification can reduce the resources required to produce a quality product if an organisation already has a systematic development approach.

As a result of the certification process, Aspect, and to a lesser extent MITS, reported significant improvements in the efficiency with which the software product was produced. Respondents reported that because certification placed emphasis on understanding customer needs, software was more accurately designed in the earlier stages of the project. This eliminated the problem of getting to the testing stage and then finding that the software which had been developed was not the software that the customer wanted. In addition, because design documents became more detailed, the number of programming errors was reduced. The increased amount of documentation in the form of comments in the code meant that errors could be located and fixed more rapidly. Overall, by cutting down on the need to return to earlier stages of the software development process, the amount of resources required to produce the software product was reduced. Like Tan and Yap (1994: 577), both Aspect and MITS found a process-oriented approach enabled errors to be detected, prevented and controlled.

Conclusion 1.3: The outcomes of certification can increase customer satisfaction.

The perception of respondents at MITS and Aspect was that certification itself - the "five red ticks" - had little impact on client satisfaction, apart from indicating that the organisation took the quality issue seriously. However, the respondents felt that the outcomes of following an effective methodology - understanding the customer's requirements and the customer's business, delivering systems on time, presenting an organised, systematic approach to the task and being able to respond quickly to requests - enhanced the customer's level of satisfaction with the product and the service. This perception - that improved project management would raise customer satisfaction, and that certification of the quality management system was of secondary importance - was also reported by Robinson and Simmons (1996). From another perspective, respondents observed that when audits are carried out on a project at a client site, the client company is necessarily involved, and to some extent, gains a free assessment of its own work habits. With the increased interaction, clients have a greater understanding of the complexity of developing a software product, and the need to follow a disciplined approach if a quality product with reduced maintenance requirements is to be achieved. Customers, at the end of the day, feel they are getting better value for money over the life of the product.

How Certification Leads to Benefits being Achieved

Certification in itself does not automatically result in benefits. Rather, it is the activities, the re-organisation and potentially the culture change that firms go through in order to meet the requirements of certification that are responsible for any positive outcomes. This section describes the mechanics by which benefits are achieved.

Conclusion 2.1: Certification allows for greater focus on the end product and increased responsiveness to customer needs.

Respondents at MITS and Aspect reported more focus on the end product, rather than less, as a result of the certification process. This contradicts Zampetakis (1994: 28) who described the certification process as a '...paper-based exercise focusing on procedure rather than end product'. At MITS, developing a formalised methodology meant that staff could focus on the task in hand, rather than wondering which task to do next. Since PWU and Aspect already had a formalised methodology prior to certification, they noticed little improvement in this area. However, all three organisations reported that having to complete detailed design documents required staff to increase consideration of customer needs. Beckworth and Altman (1996) point out that it is now widely recognised that getting these 'soft' factors right directly leads to quantum improvement in organisational performance in terms of meeting client expectations. The experience of the

case studies also shows that gaining customer feedback at other stages of the development process helps the software developer to continuously fine-tune the product to meet customer requirements.

Conclusion 2.2: Certification can improve the discipline with which professional work practices are applied, and can be used to effect cultural change through altering work behaviour and then reinforcing the new behaviour through on-going audits.

Interviews revealed that all respondents were aware of the importance of system and process documentation, and believed that this was true of their colleagues. However, in the pressured environment of delivery deadlines, documentation is often deferred (and sometimes omitted). Rouse and Watson (1995) and Dunn (in Schulmeyer and McManus 1992) point out that software is not developed in a vacuum. Factors beyond the process itself affect the scheduling of the project and, in the urge to complete the project on time, it may happen that some steps are not carried out 'by the book' - particularly those which may appear to system developers as bureaucratic paperwork imposed by management. Respondents reported that the knowledge that procedures and documentation will be audited, and that non-conformances can jeopardise certification, acts as a powerful incentive to pay attention to these less-glamorous aspects of software development. The regular audits thus provide the reinforcement for staff to practise new habits until they become second nature. Certification also acts as a catalyst for cultural change by defining work practices and specific goals. Written procedures then ensure that all staff across the certified organisation are aware of the appropriate work behaviour. Thus sound software development practices become the norm for every individual within the organisation - a universally accepted behaviour - rather than one belonging to particular individuals. When the new work behaviour has been accepted by the majority of individuals as 'the way things are done around here', then cultural change has been achieved.

Through defining processes, certification can establish new work practices; audits then provide the incentive for following the new practices. However, these will only become ingrained in the organisation if management constructs and maintains "a climate in which quality is fostered and nurtured...one where processes which facilitate quality are enabled..." (Glass, 1991:148). Without this supportive management framework, positive cultural change will not be realised.

Conclusion 2.3: Certification can improve budgeting and monitoring and increase reliability of delivery times.

Schulmeyer (1990) believes that focusing on the process through which systems are developed and delivered can help to resolve the crisis of overblown budgets and missed schedules. Similarly, Tan and Yap (1994: 577) point out that '...process management and technologies may provide the impetus to resolving the software crisis of missed schedules, overrun budgets, software defects and increasing backlog vis-a-vis applications demand'. Our case studies support these views. Aspect and MITS both agreed that the process of certification led to improved budgeting and monitoring, and more accurate schedules. Projects could be tracked more easily, and hence problems could be detected earlier in the development process. A decision could then be made whether additional resources would be required in order to get the project completed by its deadline. Early identification of problems then reduced the amount of rework required. In addition, if a task required the efforts of more than one work group, this task would be co-ordinated more effectively. This resulted in faster completion times for individual programs, which in turn enabled the more effective co-ordination of delivery times of overall projects.

It is not clear whether delivery times were reduced as a result of certification. Although there was less time spent in rework, sometimes more time was spent in the earlier stages setting up the project and determining customer requirements. However, it was reported that the discipline imposed by certification certainly led to more reliable schedules.

Overall, the experience of the our case studies support that of a previous study by Robinson and Simmons (1996) who found that internal developers perceived a quality management system as being of value in improving systems development life cycle management.

Conclusion 2.4: Certification can improve the use of human resources.

Our case studies revealed that a pre-existing software development methodology promotes a number of benefits related to the use of an organisation's human resources. Firstly, it provides a framework so that staff do not have to waste time 're-inventing' the development process. This was also argued by Perry (1992: 52), who stated that by using standards '...each individual analyst and programmer does not have to perform the task of developing the process to implement the specification. The standards are the process used'.

A pre-existing software development methodology also helps new staff to gain a knowledge of the organisation's methods and quickly absorb the way the organisation works. Staff are able to move more readily between projects since similar methods are used at different sites, and, because a paper trail exists to explain earlier decision about products and the project, staff can come more rapidly 'up to speed'. This was also described by Sanders and Curren (1995: 131) who found that a '...defined software process...allows [staff] to be moved quickly and easily between projects'.

In addition, the increased scrutiny, through audits and walkthroughs, placed demands on staff to explain what they were doing, and why. Although the audits were not specifically aimed at evaluating staff performance, they did alert project managers to the real level of staff skill and knowledge. With this information, project managers could more carefully assign their staff to appropriate tasks, with experienced staff being assigned to core elements of a project and less experienced staff being put onto less critical components. This also lead to the identification of training needs, which enabled the organisations to better target staff development programmes. This ties in with the element of certification that explicitly requires management to consider if staff have the training necessary to perform particular work functions.

Limitations of Software Quality Certification

Certification is not a silver bullet. For managers looking to certification to improve sales or the software development process in their organisations, the following limitations should be kept in mind.

Conclusion 3.1: Certification, in itself, is unlikely to lead to competitive advantage.

Respondents felt that software purchasers look at a number of factors when evaluating software systems, of which certification appears to be just one consideration. One respondent felt that word-of-mouth recommendation through prior customers was the key to winning work. Thus while certification may add to an organisation's reputation, and, for purchasers wanting products which are high in risk, cost and complexity, it may be a hurdle requirement, in general, purchasers do not appear to accept it on face value as proof of quality. Organisations without certification can still be regarded as significant competitors. On the other hand, most respondents felt their organisation faced pressure to gain certification because its absence might lead to competitive disadvantage, particularly in public sector tendering

While both MITS and Aspect won new contracts after certification, it is not possible to attribute these solely to the result of having certification per se. The experience of both these organisations would support Brown (1995) who reported that commercial benefit is not necessarily associated with certification.

Tan and Yap (1994: 583) suggested that gaining certification would '...increase the company's reputation that in turn would increase market share'. The perceptions of our respondents indirectly supported this argument. The case studies suggest that where existing clients could see that the developers were doing a better job (as a result of the visible and systematic approach to software development tasks) this led to greater customer satisfaction, which in turn enhanced the reputation of the software development organisation. This would potentially lead to improved sales. One respondent pointed out that improved efficiencies also meant that competitors could be undercut, and the savings passed on to customers.

Conclusion 3.2: Certification is not a substitute for individual skills and abilities.

While having procedures may force staff to carry out the programming in a certain way which will help make the code better, the quality of the code itself is still very much dependant on individual skills. A quality standard outlines the activities which must be documented. Documentation on software development methodology gives structure, but does not teach individuals how to program or test, nor how to carry out the other activities necessary to develop code. Individual skills are still needed to add substantive content to the structure provided by the written procedures. It is in this light that Brooks (1987), Boehm (1991), Glass (1991) and Bloor (1993) were all sceptical of the benefits of methodologies, languages and technologies in bringing about substantial improvement in software quality, and believe that the one factor which has been found to consistently improve software quality '...is the quality of the people doing the software work' (Glass, 1991: 124). The perception of respondents was that, given the same software specifications, two programmers would still come up with two different results, even using the same procedures, tools and methods. Respondents believed, though, that while a very skilled developer will produce good outputs no matter what procedures are in place, the existence of a systematic development process will bring an average developer to a higher level of performance. von Hellens (1995: 18) explains this by pointing out that 'a quality management system brings the firm to the level where software processes are repeatable. The software process becomes more visible, and, as a consequence, the successful completion is not so much dependent on the skills of the individual analyst associated with them'.

A systematic development process will also help the skilled developer produce output more efficiently. Dunn in Schulmeyer and McManus (1992: 473) supports this point by arguing that the issue of software defects can be addressed by outlining '... the employment of tools, the forms of documentation, the handover of interim products from one development group to another, configuration management, and the very methods used for problem definition and program design...'. Similar sentiments are expressed by Lee and Kim (1992: 91): '...since software development is still inclined to depend on personal skills, software development procedures must be standardized explicitly in order to enhance productivity and assure the system's quality'.

Conclusion 3.3: Registration fees are only a small proportion of the costs.

Our three case studies suggest that the cost of gaining certification is very high, and that much of this cost is hidden, as it is based on the management and staff effort that is diverted from other development activities. The visible costs such as registration and consultant fees are only a minor element of the total cost. At MITS, Aspect and PWU, these fees over three years were in the vicinity of \$12,000 and \$13,500 and \$15,000 respectively, similar to those reported by Buckler (1993) and the AIIA (1995). At Aspect and PWU, the estimated total cost of the certification effort was in the order of \$500,000, taking into account the use of external consultants, the opportunity cost of staff documenting work practices, the salaries of staff performing work, travel, administration and training. Zampetakis (1994) reported a similar cost of certification for the Melbourne office of Arthur Andersen. At MITS, no figure had been assigned to the intangible costs.

Conclusion 3.4: Certification can help improve staff and organisational performance only when the procedures themselves are perceived to be accessible and relevant.

The case studies established that certification, per se, does not result in quality improvements. Rather, it is the quality management system, and improved work practices themselves, which provide structure to the software developer, that are responsible. However, improvements don't automatically accrue from redesigned work practices. To improve outcomes, procedures for carrying these practices must be accessible and relevant. Too often, procedures sit on a shelf instead of being used (von Hellens 1995). If it is difficult or time-consuming to access the most recent version of a procedure, then staff will rely on memory rather than look up the current method. If this is the case, the benefits of process improvements will be lost, and there will be the risk of a non-conformance in a certification audit.

Problems also arise if procedures do not match the way the organisation works. In such circumstances our case studies suggest that the procedures will be ignored. For example, the initial documentation of processes

at MITS, which was written around the 21 elements of the AS 3263 standard, was long and arduous. It was only when the structure of the quality system was re-arranged to reflect the way software development was *actually* done at MITS that benefits began to accrue. Dichter (1993: 47-48) points out that 'forcing an inappropriate process on a development team will likely do more harm than good', and this was certainly the case during the initial implementation at MITS in terms of staff morale.

As well as circulating changes in procedure within the organisation, it is also important that the output from the budgeting and monitoring function is passed onto to those people who are responsible for preparing budgets or other activities which are monitored. This feedback is necessary if staff are to evaluate their work practices and further improve processes and procedures. For example, it is important that the outcome of a quotation, either successful or unsuccessful, is given back to the person who prepared the quotation. This provides the writer with feedback on whether the quote was appropriate or accurate. Too far one way could lose the client; too far the other way could cost the organisation. Either way, the person preparing the quotation needs to know the outcome in order to judge whether the criteria used in developing the quotation was appropriate. If this feedback loop is missing, the procedures regarding budgeting and monitoring can be seen to lack credibility, and may not be consistently followed. Again, the opportunity for process improvement is lost, and there is the risk of achieving a non-conformance in a certification audit.

Implementation Issues

While certification offers the potential for improvement in a number of areas, it also has the potential to create a heavy administrative burden for staff, which in turn affects the amount of time and energy left available for other software development activities. It can also reduce individual creativity and cause high levels of stress. To negate these adverse outcomes, it is essential that the certification process be effectively managed.

Conclusion 4.1: Certification is a mechanism which needs to be implemented in a manner consistent with sound management policies or else it can become a significant source of stress.

Respondents in both PWU and MITS reported high levels of personal and organisational stress during their initial implementation of software certification.

At PWU and MITS, having to create extensive documentation was seen, by some, to be a slur on professional ability. Individuals saw little need to document procedures 'to the nth degree', and felt that having to follow the detailed procedures was an insult to their knowledge of their jobs. Gillies (1992: 149) commented on this, reporting that '...many software developers appear to be quite content with the current state of the quality of software. They are resistant to new ideas, which are seen as a threat to their integrity and professionalism'. Humphrey (1989: 139) also pointed out that 'It is pretty hard for anyone to be objective about auditors. We generally do our own jobs pretty carefully and resent any contrary opinion'.

Another cause for resentment was the pressure applied by management to gain certification. At PWU, strong emphasis was placed on getting certification quickly because of the market imperative. Since software development methodologies were already well established, the emphasis was on ensuring that the paperwork complied with the audit requirements. This emphasis was reinforced by the requirements of the third-party auditor. At MITS, because the initial implementation was based around the 21 elements of the standard, rather than tailored to the routine software development activities, a similar situation existed. At both organisations, the paperwork, rather than the process, was seen to be the driver, and imposed significant additional workload for staff. This led to considerable staff cynicism and resentment of the certification process. Further, certification and its processes were seen as an exercise separate from the daily workings of the organisations. These two factors considerably increased the stress levels of staff. The implications of the lack of appropriate management support has been clearly expressed by McGeorge (1992: 22) who stated 'that without the appropriate supportive infrastructure (empathetic leadership, people empowerment, relationship building, behavioural retraining), standards and procedures in themselves will not bring about a culture wherein quality is the natural way to do business'.

Both these organisations found that the intensity of effort required to pass the audits left staff with little energy to maintain the system between audits. This led to a vicious circle since the same heroic effort would be required for subsequent audits.

MITS later changed its focus from the documentation to the underlying processes and, in doing so, removed much of the apparent administrative burden from staff. Administration was then seen as part of the everyday activities and an important by-product of the process, where previously it had been seen as something that had to be done on top of 'the real work'. This shift in emphasis, together with the visible process improvements, led to much greater acceptance of the certification process. In its turn, PWU found that changing to a third-party auditor that viewed the quality system in terms of continuous improvement and as an integral part of everyday processes, rather than strict adherence to certification requirements, brought about a change in staff attitude.

In contrast to the situation in these organisations, Aspect, from the start, designed the written procedures around the organisation's existing business processes. Because the written procedures were a reflection of daily work, stress was not manifested to the point where it became a major concern.

Conclusion 4.2: Certification's effect on creativity depends on the nature of the procedures and their implementation.

At MITS, the initial implementation was seen to restrict creativity within the software development activities because staff were being asked to conform to procedures which were seen to be irrelevant to everyday software development activities. In this situation, the procedures had become the driver of work practices. This was, perhaps, what Glass (1991) and Dichter (1993) respectively had in mind when talking about '...lockstep groups of marchers moving indistinguishably toward an uninteresting goal' and 'process fever'. However, once the system was implemented more sensitively, it became a process upon which staff could depend for a reliable outcome. This supports the findings of Dunn (in Schulmeyer and McManus 1992), Glass (1991) and Curtis and Paulk (1993), who all essentially argue that standardisation need not restrict artistry and individualism, but instead provides a controllable process which leads to a stable and predictable outcome. A MITS respondent felt that it was essential that the system allow for procedures to be readily changed to meet changing business requirements. Without this flexibility, the quality system, instead of supporting continuous improvement, would actually be preventing it.

At PWU, it was felt that the procedures required by the certification process were more suited to traditional development methods, and that it would be difficult to make the newer software development strategies, such as prototyping and object-oriented technologies, comply with the procedures, since these strategies were perceived to require a high degree of flexibility. Inwood (1993) and Baker and Rouse (1996) found that the use of standards would be perceived as more restrictive in organisations using newer software development methodologies (such as prototyping and object-oriented technologies) where there is less emphasis on traditional analysis, and a greater emphasis on iterative development and on-going deliverables. However, at MITS, where object-oriented technologies had been in use for some time, no differentiation was made between writing procedures for these methodologies and writing for more traditional tools. This would imply that it is the effort of having to adapt procedures to the new tools which causes the difficulty, rather than the tools themselves.

Respondents noted that to some extent whether or not documented procedures were seen as restrictive depended partly on personality, since some individuals appreciate the structure and guidance provided by work procedures, while others perceive them as a necessary evil. Encouraging independently-minded skilled individualists to work with the quality management system is a particular challenge for management (Hovenden et al, 1996).

Conclusion 4.3: The time required to gain certification is offset by the transferability of procedures and documentation across projects and sites.

Certification takes an average of eighteen months to achieve (AIIA, 1993)s. At PWU, certification was achieved in just under eight months. This was possible through the dedicated efforts of staff and management. It was also possible because many of the software development procedures at PWU were already well documented, and the emphasis was on gaining certification, rather than effecting organisational change. At MITS and Aspect certification was achieved in just under two years. Respondents at all sites reported that once documentation relating to one project had been established, it saved time because the original project plan could then be used as a template for subsequent projects. This reinforces the arguments of Perry (1992) and Sanders and Curran (1995) who point out that with written procedures providing a framework, time is saved, staff are able to adapt to new situations, and human resources are used more productively. Thus the discipline imposed by certification enhances opportunities for greater organisational learning, and consequent efficiencies.

CONCLUSION

These case studies have established that certification is a tool for quality management and improvement. Our data provides strong evidence that used appropriately, certification can act as a catalyst for change, and a reinforcement for quality management practices that may get given lower priority in its absence. However, used inappropriately, the certification process can cause considerable costs to an organisation and its workforce.

While benefits were achieved in a number of different areas in our case studies, the experiences of the organisations suggest that benefits are more likely to be realised under certain conditions:

- First, benefits are more likely to be realised if the certification process is implemented using sound management techniques such as open communication, provision of resources and training, evident management commitment and employee participation. The initial, and then later experience at MITS, shows that implementing certification without careful attention to business objectives as well as the requirements of the standard is likely to result in further organisational costs, rather than benefits.
- Second, the certification process is likely to be of greatest value to organisations lacking formalised development procedures and to those that feel that software development is not performing to the necessary standard. In such situations the institution of the systematic approach to software development required to gain certification is likely to bring about a change in software quality. Thus MITS, which lacked its own formal development methodology, noticed a distinct change in the final software product as a result of its quality management system. But PWU and Aspect, which already had formalised development methodologies, noticed little change in the final product.

Our studies have also indicated that even where organisations already have a systematic approach to software development in place, certification can still result in benefits by reducing the resources necessary to arrive at the desired product.

- Third, a systematic approach to software development is not a substitute for individual skills and abilities. This was emphasised by members of all organisations. Work procedures enable individual staff members to be more productive, but are of little use if an organisation lacks individuals with the skills necessary to perform the development work.
- Fourth, the quality system will be only as good as the work practices it documents, and the accessibility of this documentation. This was demonstrated at MITS when the written procedures in relation to budgeting and monitoring described the process of preparing job quotations, but the system did not require the outcome of the quotation to be fed back to the writer. In addition, if staff found it difficult to download latest versions of procedures, they tended to use the procedures on hand, or else rely on

memory. Our studies emphasise that procedures should be readily accessible, and that it should be easy for individuals to track down relevant changes to the procedures so time is not wasted, which in turn can engender resentment against the quality system.

These case studies provide strong evidence of certification's potential to create positive outcomes for software development organisations. The benefits an organisation gains from certification will vary, depending on the organisation's reasons for gaining certification, the attitude of management and staff, the degree of formalisation of software development methods, the extent of existing documentation, and the accessibility and relevance of the documentation generally. Certification offers a way for managers to systematically bring about positive change to an organisation, but like any tool, it must be appropriately wielded if the desired outcomes are to be achieved.

REFERENCES

- AIIA: Australian Information Industries Association (1993), *Quality Management Issues: An IT Industry Perspective*, AIIA, Canberra.
- AIIA: Australian Information Industry Association (1995), *Membership Quality Survey Report*, AIIA, Canberra.
- Baker, M. (1996), "Quality Certification: Management Strategies for Success", *Proceedings from the Academy of International Business South-East Asia Region*, Dunedin, pp. 26-30.
- Baker, M., and Rouse, A. (1996a), "Software Quality Certification: Overcoming the Obstacles", *Australian Journal of Information Systems*, Vol 3 No 2, pp. 3-9.
- Baker, M., and Rouse, A. (1996b), "Effects of Certification on Selected Aspects of Software Development", *Proceedings from the 7th Australasian Conference on Information Systems*, University of Tasmania, December, pp. 45-56.
- Beckworth, G. and Altmann, G (1996), 'Defining strategies - measuring quality', *Software Quality Journal*, Vol 6 No 2, pp 171-178.
- Bloor, R. (1993), 'Software quality standards; do software quality initiatives, such as ISO 9000, reduce the introduction of dangerous programmable errors?', *DBMS*, Vol 6 No 2, pp. 10-11.
- Boehm, B.W. (1981), *Software Engineering Economics*, Prentice-Hall Inc, New Jersey.
- Brooks, F.P. Jr. (1995), *The Mythical Man-Month*, (2nd ed), Addison-Wesley, Reading.
- Brown, A. (1995), 'Industry Experience with ISO 9000', *Proceedings - 2nd National Research Conference on Quality Management*, pp.147-159, Perth.
- Buckler, G. (1993) ""ISO" designation shows quality a priority", *Computing Canada*, Vol 19 No 18, p. 8.
- Bush, M. (1993), "Incremental quality improvements in the software development process and the first levels of the SEI Capability Maturity Model: A view from inside the whale". *Proceedings from the 2nd International Conference on Achieving Quality in Software*, Italy, pp. 215-223.
- Curtis, B. and Paulk, M. (1993), "Creating a software process improvement program", *Information and Software Technology*, Vol 35 No 6/7, pp. 381-386.
- Deming, W.E (1986), *Out of the Crisis*, MIT Centre for Advanced Engineering Study, Massachusetts.
- Dichter, C. (1993), 'Software Audits', *Unix Review*, October. pp. 43-49.

- Dunn, R.H. (1992), The Quest for Software Reliability in *Handbook of Software Quality Assurance*, (2nd ed.), (eds) Schulmeyer, G.G. and McManus, J.I., Van Nostrand Reinhold, New York, pp. 465-502.
- Gillies, A.C. (1992), *Software Quality: Theory and Management*, Chapman & Hall, London.
- Glass, R.L. (1991), *Software Conflict: Essays on the Art and Science of Software Engineering*, Yourdon Press, New Jersey.
- Hovenden, F.M., Walker, S.D., Sharp H.C., and Woodman, M. (1996) 'Building quality into scientific software' *Software Quality Journal*, Vol 5 No 1 pp. 25-32.
- Hoyle, D. (1994), *Quality Systems Handbook*, Butterworth/Heinemann, Oxford.
- Humphrey, W.S. (1989), *Managing the Software Process*, Addison-Wesley Publishing Company, Massachusetts.
- Inwood, C. (1993), "We Can't Escape the ISO Certification Push", *Computing Canada*, Vol 19 No 9, p. 25.
- Lee, J. and Kim, S. (1992), "The relationship between procedural formalization in MIS development and MIS success: A contingent analysis", *Information and Management*, Vol 22, pp. 89-111.
- McGeorge, B. (1992), 'Quality Software', *Quality Magazine*, August, pp. 21-23.
- Perry, W. E (1992), "Quality Concerns in Software Development", *Information Systems Management*, Vol 9 No 2, pp. 48-52.
- Robinson, K and Simmons, P. (1996), "The value of a certified quality management system: the perception of internal developers", *Software Quality Journal*, Vol 5 No 2, pp 61-73.
- Rouse, A.C. and Watson, D.J (1995), 'Applying TQM to Information Systems Quality: The Role of Culture', *Asia Pacific Journal of Quality Management*, Vol 4 No 1, pp.12-23.
- Ryan, I. (1994), "Caltex taps the good oil on quality", *Managing Information Systems*, June, pp. 63-65.
- Sanders, J. and Curran, E. (1994), *Software Quality*, Addison-Wesley Publishing Company, Reading.
- Schulmeyer, G.G. (1990), *Zero Defect Software*, McGraw Hill, New York.
- Schulmeyer, G.G. (1992), Software Quality Lessons from the Experts in *Handbook of Software Quality Assurance*, (2nd ed.), (eds) Schulmeyer, G.G. and McManus, J.I., Van Nostrand Reinhold, New York, pp. 25-48.
- Tan, M., and Yap. C. Y. (1994), "A Study of TQM Practice on Software Development in Singapore", *Proceedings from IFIP Working group 8: Conference on Business Process Re-Engineering*, Gold Coast, pp. 575-484.
- von Hellens, L. A. (1995), "Quality Management Systems in Australian Software Houses: Some Problems of Sustaining Creativity in the Software Process", *Australian Journal of Information Systems*, September, pp. 14-24.
- Zampetakis, H. (1994), "Investment in quality accreditation queried". *Australian Financial Review*, January 31, p. 28.

ACKNOWLEDGMENTS

The authors thank the staff of MITS, Aspect Computing, Price Waterhouse Urwick, and Standards Australia for their generous assistance in these case studies.

COPYRIGHT

Megan Baker and Anne Rouse © 1997

¹In the future this will be sought in conjunction with the recently developed guideline AS/NZS 3905.8: 'Quality systems guidelines Part 8: guide to AS/NZS ISO 9001:1994 for the software industry'. The current standard AS 3563: *Software Quality Management System* will become obsolete in June 1998.

²While all of PWU had gained certification to ISO 9001, this study was confined to PWU's information technology management consulting services group (MCS) which had also gained certification to AS 3563.