December 1993

Managing Software Quality Assurance

Kin-Lee Seah
National University of Singapure

Chee-Sing Yap
National University of Singapure

MANAGING SOFTWARE QUALITY ASSURANCE

Kin Lee SEAH, Department of Information Systems and Computer Science
National University of Singapore, Singapore 0511
(internet: seahkinl@iscs.nus.sg)

Chee Sing YAP, Department of Information Systems and Computer Science
National University of Singapore, Singapore 0511
(internet: yapcs@iscs.nus.sg)

ABSTRACT

The Total Quality Management movement in recent years has focused management attention on the issue of quality. Managers are now generally much more aware of the importance of quality assurance and quality control. In information systems, software quality assurance has become a key management issue, mainly because of an increased awareness of the importance of software quality and cost consideration. The cost of the software component of any computer-based solution is more than 80% of the total cost. The cost of detecting and correcting errors in software increases almost geometrically during the software life-cycle. A properly managed software quality assurance programme will significantly reduce the number of errors and ensure that these are detected early in the software development life cycle, hence reducing software cost significantly. This paper reviews the literature on software quality assurance, provides guidelines on management of software quality assurance programmes, and discusses issues relating to policies, staffing, structuring, education, resistance to change, and the software quality management maturity grid.

NEED FOR SOFTWARE QUALITY ASSURANCE

Software Quality Assurance (SQA) is undoubtedly one of the key issues facing most Information Systems (IS) organisations today, regardless of whether it is an internal IS department or a software vendor providing software solutions in the Information Technology (IT) market. The cost of the software component of any computer based solution is more than 80% of the total cost [Boehm, 1981]. Cost of detecting and correcting errors in software increases almost geometrically during the software life-cycle [Vincent, 1988]. A properly managed SQA program will significantly reduce the number of errors and ensure that these are detected early in the software development life cycle, hence reducing software cost significantly. For the software vendors, a highly competitive environment demands a better quality product at the existing price or at a lower price [Edosomwan, 1987; Grady, 1987]. Customers demand for assurance of software quality through software certification has also led to the development of SQA standards and numerous institutes offering software certification. Overzealous SQA methodology consultants are always ready to "sell" their methods. Management must be aware of their roles and responsibilities, the required fundamentals for a successful SQA program, and be able to make preliminary assessment on the readiness of the organisation in seeking certification. These would greatly enhance the success of the SQA program and avoid the high cost in a failed SQA program or certification attempt.

WHAT IS SOFTWARE QUALITY ASSURANCE

Crosby [1986] defined quality as "conformance to requirements". Vincent [1988] defined assurance as "the process by which we, with surety, achieve the desired goal of quality". In the IEEE Standard Glossary of Software Engineering Terminology, software quality is defined as "the degree to which software possesses a desired combination of attributes". In the following discussion, we will use the following definition:

> "Software Quality Assurance is a planned and systematic pattern of all actions necessary to provide adequate confidence that the software product conforms to established requirements."[1]

The phrase *"planned and systematic pattern of all actions"* itself suggests a certain level of managerial actions required to achieve the definition.

MANAGEMENT RESPONSIBILITY FOR SQA

Juran [1989] divided quality management into a hierarchy of "strategic quality management", "operational quality management", and "the work force and quality". In this paper, the discussion will focus mainly on strategic quality management for the software development process which Juran emphasized as the centroid to the entire subject of making quality happen, all else being supportive. The Quality Assurance Institute [1987] listed management responsibility as one of the three key principles of SQA and the ISO [ISO9004, 1987] stated that "the responsibility for and commitment for a quality policy belongs to the highest level of management".

Consideration of Risks, Costs and Benefits

The first responsibility for any IS manager intending to implement a SQA program is to evaluate it in terms of risks, costs, and benefits from the company and customer perspective [ISO9004, 1987]. The risks related to deficient product for the company include loss of market, claims, liability, and loss of image or reputation. Consideration has to be given to risks for the customer in terms of health and safety of customer, availability, and loss of confidence. The cost dimension for the company will include cost of fixing errors, rework, and cost of implementation of the SQA program while the consumer could potentially incur cost in downtime and reacquisition/relearning. Major benefits for the company would be increased profitability and market share. The customer could also gain in terms of reduced cost and improved fitness for use.

[1]Adapted from IEEE software engineering terminology and DOD Directive 4155.1, Quality Assurance, 9 February 1972, as reported by Schulmeyer, G. G. in the book "Handbook of Software Quality Assurance", Van Nostrand Reinhold Company, New York, 1987.

Management Commitment

Undoubtedly, management commitment is the single most critical step to achieving successful software quality [Cavano, 1987; Crosby, 1986; Edosomwan, 1987; Dobbins, 1987; Quality Assurance Institute, 1987]. An EDP Quality Assurance Survey conducted by the Quality Assurance Institute [Perry, 1987] in 1983 identified it as the most serious problem associated with the practice of SQA. The proper attitudes, organisational structure, policies, and reward system are required to make software quality happen. These cannot occur without the wholehearted support and involvement of the management. Tight schedule and insufficient funds are frequent occurrences during software development. IS managers' commitment to software quality will ensure that faced with the three managerial tradeoff decision between quality, schedule, and budget, quality will be the first and only choice[2].

Quality Policies and Objectives

Management should develop and state its software quality policy and objective. Where IS is a function in the organisation, this policy should be consistent with other corporate policies. The SQA policy has the dual function of being a formal statement of the organisation's commitment to quality software [Quality Assurance Institute, 1987] and letting its employee knows the key elements of its implementation [Crosby, 1986]. The key responsibilities of management in the policy formation process includes helping to identify areas that the software policy should address, assigning responsibility for drafting the policy statements, and reviewing and approving the policy [Juran, 1989].

Policies established must be disseminated throughout the organisation to be understood, implemented, and maintained. Effective communication is critical to ensure that the company reaches out to everyone and that they understand and recognise their role in causing software quality to happen [Edosomwan, 1987; Vincent, 1988].

Resources and Organisational Infrastructure

Implementing the list of tasks to achieve software quality requires organisational resources and supporting organisational infrastructure. A major failing of upper managers has been the failure to provide the necessary resources to make the SQA program effective [Juran, 1989]. Such a failure sends a negative message to the lower levels.

Major software quality problems are likely to be interdepartmental in nature, even where IS is a support function in the organisation. Hence, interdepartmental teams are needed. Such teams require legitimacy, priorities, and resources that are best provided from upper-management sources [Juran, 1989].

At the highest level, a quality council could be established. The council shall consist of a chairperson and a staff of quality coordinators drawn from the major systems divisions in the company. Each coordinator will be responsible for coordinating software quality program in their division. For the internal IS department, members should include user management to ensure that users are consulted on major software quality issues and they could provide feedback on IS quality level. The quality council could also act as an effective channel for communicating software quality policy.

Recognition and Reward System

Quality work should be rewarded by management through several means, such as awards, recognition among peers, bonus pay, additional technical challenge, and promotion. If the goals are revised (with new emphasis or goals on quality) but the reward system is not, the result as viewed by subordinates is conflicting signals. Most subordinates would resolve this conflict by following the priorities indicated by the reward system [Juran, 1989]. The reward system for quality work should be built into the organisation's formal appraisal system so that it is institutionalised and software quality becomes an objective of the employee.

Staffing

To be effective and to contribute to the organisation's SQA program, the SQA function must be staffed by competent professionals. Quality assurance within the data processing profession is a relatively new function. It also differs significantly from the engineering/manufacturing quality assurance in: first, IS products and services are unique products, making it a job-shop function, as opposed to a continuous process; and second, data processing is a high-technology area undergoing continuous rapid change [Quality Assurance Institute, 1987]. IS managers hence find great difficulties in formulating job description and evaluation of personnel to recruit for the SQA function.

The Quality Assurance Institute [1987], in a research project, identified thirteen generic skills[3] as important to the practice of SQA. IS managers could make use of these as guidelines for recruitment or rely on formal qualification. In the staffing process, IS managers should look into factors like: is it possible to promote from within and train individuals to fill the openings and the likelihood of getting the right candidates at the local, regional or international level [Mendis, 1987].

It is essential that SQA professionals in the organisation has the opportunity to ascend the corporate ladder. This is particularly important for the internal IS support department where the SQA function is likely to be smaller. It is imperative that management recognises this critical shortfall and takes steps to remedy it in order to attract SQA professionals and develop the SQA function. Management should plan and define career paths from SQA to other disciplines within the organisation, such as from SQA to software development management or general management. IS managers should develop an appraisal system for the SQA professionals, taking into consideration the current organisational appraisal system.

Training/Education

Extensive education at all levels of the organisation is required to make the SQA program operative and effective [Crosby, 1986;

[2]It may seem that the 3 goals of quality, cost, and schedule are conflicting and mutually exclusive. It is not true. Significant improvements in both cost and schedule can be achieved as a result of focusing on quality [Walter, 1983].

[3]The 13 skills are : 1) Communications, 2) Training Development and Delivery, 3) Principles of quality, quality control, and quality assurance, 4) Principles of marketing, 5) Analytical (statistical) problem-solving and report, 6) Verification and validation, 7) Policies, standards, procedures development and administration, 8) Leadership, 9) Principles of auditing and control, 10)Principles of data processing, 11) Project management and control, 12) Financial analysis, 13) Principles of change. Each skill is accompanied by a description of the skill and the objective for including it.

Deutsch, 1988; Edosomwan, 1987; Ishikawa, 1985; Juran, 1989]. Organisation should recognise that this requires an investment of time and money. Juran [1989] recommended that different training package should be used for employees in different organisational levels. At the senior management level, it should concentrate on the role of managers in managing for quality. Operational management should be trained in the planning and review of SQA program. Training at the workforce level would aim to arm them with specific tools and methodology of SQA.

Education program could be conducted in-house or by external parties. Companies that wish to conduct in-house training but lack the expertise could invite external consultants to develop training packages and conduct initial lessons. Expertise is then gradually transferred to in-house SQA trainers.

Ishikawa [1985] warned against the pitfall of not combining personnel placement and organisation plans with the SQA education plans. If these two closely related areas are uncoordinated, it could result in the wastage of substantial education cost and the unavailability of trained personnel to implement the SQA function.

Review

Review of progress is an essential part of assuring that goals are being met. This could be done by having summarised progress reports of the SQA program and comparing results with milestone targets. Another complimentary method is the use of software quality audits. Management review also sends a message to the organisation on the priority and importance of quality.

ROADBLOCKS OF SOFTWARE QUALITY ASSURANCE

Management commitment, SQA policies and objectives, provision of resources, staffing of SQA professionals and education are the necessary conditions for a successful SQA program, but it is not sufficient. Management should be aware of the major hurdles along the way and take necessary actions to overcome them.

Resistance to Change

The IS department is not unfamiliar with user resistance during system implementation [Markus, 1983]. SQA faces the same problem in the IS function [Edosomwan, 1987; Juran, 1989]. This could result from the effect of change on the status, beliefs, practices, and habits of those affected. A good basic strategy to minimise resistance is to get everyone involved in all phases of the project and SQA program when possible, avoid surprises, create a favourable social climate, and solicit feedback and respond to them positively [Edosomwan, 1987; Juran, 1989].

Fear of the Unknown

Although SQA is not totally new and has been used extensively and tested in several pioneer software organisations, it still has unproven results. Most people are likely to be afraid of failure and as a result, they may resist the implementation of the new concepts and ideas. All levels of management and employees must be educated about the essentials and importance of willingness to risk failure [Edosomwan, 1987]

Resentment of Criticism

Most people believe that they perform their task in the most effective and efficient manner and will resist criticism that proves otherwise, which is part of the SQA process and unavoidable [Edosomwan, 1987]. Criticism should be offered constructively and personnel at all levels must have an open mind and have the willingness to accommodate several different viewpoints. Management should lead by example and create an organisational culture where constructive criticism and willingness to accept it are the norms.

STRUCTURING FOR SQA

Several organisational structures can be applied to any SQA program. An important concept is the determination of its role as either an operating department in the line organisation or as a staff group reporting to top management. The size of the IS group will also constraint the various options available [Wasmuth, 1986]. Smaller IS groups, especially those operating as a support function in the organisation, are likely to adopt a staff position (reporting to the IS manager) for the SQA function. A staff group has the ability to move freely outside the operating environment but the major disadvantage is that the group is often not empowered to do other than report the situation [Wasmuth, 1986]. Line organisation allows SQA professionals to build better rapport with the other IS professionals and in a highly distributed environment, it ensures that SQA expertise is readily available. Where the IS group is an internal department, the relationship of the SQA group to the corporate quality group and the line of responsibility should be clearly specified. The organisation of the SQA group will also impact the career paths of the SQA professionals. Management should be aware of all these issues and take into consideration the overall organisational fit in the current structure when structuring the SQA function.

SOFTWARE QUALITY MANAGEMENT MATURITY GRID

Philip Crosby [1979] in his book *Quality is Free* developed the Quality Management Maturity Grid. Schulmeyer [1987] examined it in the context of software production and modifie it to become the Software Quality Management Maturity Grid shown below (Figure 1).

Awareness of the different stages allows management to assess the organisation's current stage and to take proactive actions to prepare the organisation for an accelerated move to the next stage.

Schulmeyer proposed his model as a straight linear stream of stages. Although the mainstream corporate software development environment will move from one stage to another linearly, "sub-stream" could develop as the organisation has to cope with new technology. McFarlan [1983] proposed a four phases model of technology assimilation consisting of: 1) Phase 1 - Identification and initial investment, 2) Phase 2 - Experimentation & learning, 3) Phase 3 - Control, 4) Phase 4 - Widespread technology transfer. Although SQA techniques and methods are generic, there are specific standards and methods for different technology. For example, the traditional way of testing the system output with expected result using exact Boolean match may not be applicable for an expert system where result could not be measured by exact match but by relative closeness. Organisations at the first two phases of technology assimilation would lack the understanding and specific standards applicable that is required for writing a SQA Plan. They would start at the enlightenment stage and move up at an accelerated pace to join the mainstream on acquiring the required techniques.

For organisations in the first two phases of technology

STAGES	CHARACTERISTICS
Stage 1 **UNCERTAINTY**	There are five quality "facts" that software development believes: 1. Quality means goodness, it cannot be defined. 2. Because it cannot be defined, quality cannot be measured. 3. The trouble with quality is that American workers don't give a damn. 4. Quality is fine, but we can't afford it. 5. Data Processing is different - errors are inevitable.
Stage 2 **AWAKENING**	SQA is called upon in crisis situations.
Stage 3 **ENLIGHTENMENT**	The SQA Plan is written first as the "driver" to the software development effort.
Stage 4 **WISDOM**	SQA management and software development management are working together to produce quality software.
Stage 5 **CERTAINTY**	Quality software is produced on time within cost every time.

Figure 1 - Software Quality Management Maturity Grid [Schulmeyer, 1987]

assimilation, users and IS professionals need some freedom to familiarise themselves with the technology and errors are inevitably higher. IS management must be aware that attempting to enforce SQA program too strictly for new technology at the first two phases of assimilation could backfire and stem further growth in the use of the technology.

CONCLUSION

To remain competitive, software companies will need to continue producing higher quality software and seek certification. Experienced users would also demand higher quality software from the internal IS department. In both situations, the solution lies in an effective and efficient SQA program. Management and IS managers must understand the basic issues involved. SQA is a new and increasingly important responsibility for both management and IS managers.

References

Boehm, B.W. Software Engineering Economics, Prentice-Hall, Englewood Cliffs, N.J., 1981.

Cavano, J. P. and LaMonica, F. S. (1987) "Quality Assurance in Future Development Environments", IEEE Software, pp. 26-34, September 1987.

Crosby, P. B. "Management and Policy" in Quality Management Handbook edited by Walsh, L., Wurster, R. and Kimber, R. J., Marcel Dekker Inc, New York, 1986.

Crosby, P. B. Quality without Tears: The Art of Hassle-Free Management, McGraw-Hill, Singapore, 1986.

Crosby, P. B. Quality is Free, McGraw-Hill, New York, 1979.

Deutsch, M. S. and Willis, R. R. Software Quality Engineering: A Total Technical and Management Approach, Prentice Hall, Englewood Cliffs, 1988.

Dobbins, J. H. and Buck, R. D. "The Cost of Software Quality" in Handbook of Software Quality Assurance edited by Schulmeyer, G. G. and McManus, J. I., Van Nostrand Reinhold Company, New York, pp. 119-136, 1987.

Edonsomwan, J. A. Integrating Productivity and Quality Management, Marcel Dekker Inc, New York, 1987.

Grady, R. B. "Measuring and Managing Software Maintenance", IEEE Software, pp. 35-45, September 1987.

Ishikawa. K. What is Total Quality Control? The Japanese Way, translated by Lu, D. L., Prentice Hall, Englewood Cliffs, 1985.

ISO9004 Quality Systems Part 4 : Guide to Quality Management and Quality System Elements, International Standards Organisations, 1987.

Juran, J. M. Juran on Leadership for Quality: An Executive Handbook, The Free Press, New York, 1989.

Markus, M. L. "Power, Politics, and MIS implementation", Communications of the ACM, Vol. 26, No. 6, pp. 430-444, 1983, 1983.

McFarlan, F. W.; McKenney, J. L. and Pyburn, P. "The Information Archipelago - Plotting a Course", Harvard Business Review, Jan - Feb, pp. 145-156, 1983

Mendis, K. S. "Personnel Requirements to Make Software Quality Assurance Work", in Handbook of Software Quality Assurance edited by Schulmeyer, G. G. and McManus, J. I., Van Nostrand Reinhold Company, New York, pp. 104-118, 1987.

Perry, W. E. "Effective Methods of EDP Quality Assurance", in Handbook of Software Quality Assurance edited by Schulmeyer, G. G. and McManus, J. I., Van Nostrand Reinhold Company, New York, pp. 408-428, 1987.

Quality Assurance Institue Report on Data Processing Quality Assurance: Charter, Skills, and Job Descriptions, Quality Assurance Institue, Florida, 1987.

Schulmeyer, G. G. "Software Quality Lessons from the Quality Experts", in Handbook of Software Quality Assurance edited by Schulmeyer, G. G. and McManus, J. I., Van Nostrand Reinhold Company, New York, pp. 25-44, 1987.

Vincent, J.; Waters, A. and Sinclair, J. Software Quality Assurance (Volumen 1): Practice and Implementation, Prentice-Hall, Englewood Cliffs, N.J., 1988.

Walter, C. "Management Commitment to Quality: Hewlett-Packard Company", Quality Progress, Aug 1983.

Wasmuth, K. F. "Organisation and Planning", in Quality Management Handbook edited by Walsh, L., Wurster, R. and Kimber, R. J., Marcel Dekker Inc, New York, 1986.