TOTAL QUALITY MANAGEMENT, VOL. 9, NO. 7, 1998, 611–618

CARFAX

Analysis of the total quality management-based software auditing

DAVID C. CHOU[1], DAVID C. YEN[2] & JIM Q. CHEN[1]
[1]*Department of Business Computer Information Systems, St Cloud State University, St Cloud, MN 56301-4498, USA &* [2]*Department of Decision Sciences & MIS, Miami University, Oxford, OH 45056, USA*

Summary *As the total quality management (TQM) method becomes the mainstream of quality control, a new auditing framework is needed to guide the process of software quality assurance. This paper discusses the importance of the TQM method and proposes a TQM-based information system auditing framework. A TQM-based quality auditing could enhance the e ectiveness and efficiency of the software quality management.*

Introduction

Software quality is an important factor in gaining competitive advantages in the software industry. Software users or customers ask for the commitment of providing quality software by software development companies. Software shops eagerly search for a suitable software development process in order to meet their user or customer needs.

Total quality management (TQM) is a paradigm that guides the process of organizational management with a 'customer-oriented' philosophy. TQM is a systematic mechanism in which an organization is treated as an interactive network fulfilled with communication and control. Quality management must ensure that horizontal integration across networks and vertical integration through hierarchies are achieved (Flood, 1993).

TQM can be divided into the following processes: plan, do, check and act. These processes are often referred to collectively as the P–N–C–A cycle (Arter, 1994). 'Plan' is the preparation process. 'Do' is the action as planned. 'Check' is the measurement activity, including inspection, surveillance, audit, appraisal, evaluation and review. 'Act' is the improvement stage after problem detection. The TQM method can be used to conduct the software quality assurance (SQA) process. The SQA process is a technique that achieves software quality through the software engineering process. The SQA process includes activities such as adopting analysis, design, coding and testing methods and tools; applying formal technical reviews during each software engineering step; using a multi-tiered testing strategy; controlling software documentation and the changes made to it; delivering a procedure to ensure compliance with software development standards; and utilizing measurement and reporting mechanism (Chou, 1994a).

Another important SQA process is software quality auditing. Software quality auditing is a process that determines whether various elements within a quality system meet their stated quality objectives (Chou, 1996). Traditionally, auditing deals with the last two steps

0954–4127/98/070611-08 $7.00

in TQM's P–D–C–A cycle, that is, it only performs the check and act functions in the quality cycle (Arter, 1994). However, this concept could be challenged by employing a proactive and integrated software quality auditing framework.

Software quality auditing should monitor the whole system software development life cycle (SDLC) through a strong communication and coordination mechanism at the workplace. This process involves customers, systems analysts, customer service personnel, auditor and project manager at the beginning of the SDLC. The auditor should perform an active role in identifying the standards and objectives of information systems (ISs) at a joint application development (JAD) session. He or she will then monitor the construction processes of the ISs throughout the SDLC. This new approach integrates the IS auditing function into a TQM environment for enhancing the SQA process.

This paper first discusses the concepts of the IS auditing process and the TQM methodology. Some weaknesses of existing IS auditing processes are then identified. A new IS auditing process is proposed with its framework and the implementing process analysed in the latter sections.

SQA and the software auditing process

SQA process

Software quality was defined by Pressman (1992) as: "Conformance to explicitly stated functional and performance requirements, explicitly documented standards, and implicit characteristics that are expected of all professionally developed software". This definition implies three important points: software requirements are the foundation from which quality is measured; specified standards define a set of development criteria that guide the manner in which software is engineered; and there is a set of implicit requirements that often goes unmentioned.

The goal of software quality management is to ensure that customers are satisfied with delivery. Customer satisfaction will guarantee continued contracts. Customers might want to add functionality or make other changes to reflect their business environment and needs. IS shops must be able to adapt to these frequent changes in a timely manner. However, these explicit and implicit requirements create uncertainty in the SQA process.

SQA activities include the following seven tasks (Pressman, 1992):

(1) application of technical methods;
(2) conducting formal technical reviews;
(3) software testing;
(4) enforcement of standards;
(5) control of change;
(6) measurement;
(7) record keeping and reporting.

SQA activities begin with a set of technical methods and tools that help the system analyst and designer to build up high-quality software. It includes methods and tools such as JAD sessions, structured approach, object-oriented method, quality function deployment (QFD) and computer-aided software engineering (CASE) tools. A careful use of these methods and tools can gain benefits of high-quality and production automation during system analyses, design, coding and development processes.

The quality assessment process is implemented through the formal technical review. The formal technical review is a meeting conducted by technical staff to highlight possible quality problems of a software product. Software testing includes a series of test case design methods

that help to ensure error detection. If formal standards exist, an SQA activity must be conducted to ensure that they are being followed. Standards may be dictated by customers or a regulatory mandate, although sometimes they are self-imposed by IS shops.

Measurement is an integral part of the SQA process. Various software metrics on technical and managerial measures must be collected for analytical purposes. Record keeping and reporting provide collection and dissemination of software quality information.

The best way to identify software defects is through the use of walkthroughs, reviews and inspections (Johnson, *et al.* 1995). Most SQA activities, including technical review, testing, standards enforcement, change control, measurement, and record keeping and reporting, can be fulfilled through these three methods (Chou, 1996). Auditing is another way of performing SQA.

Auditing process

One way of assessing the quality control process is through auditing. Audits should be implemented in order to determine whether various elements within a quality system meet their stated quality objectives. Quality audit is defined as (ISO, 1990): "Systematic and independent examination to determine whether quality activities and related results comply with planned arrangements and whether these arrangements are implemented effectively and are suitable to achieve objectives".

An audit is normally designed for one or more of the following purposes (ISO, 1990):

(1) to determine the conformity or non-conformity of the quality system elements with specified requirements;
(2) to determine the effectiveness of the implemented quality system in meeting specified quality objectives;
(3) to provide the auditee with an opportunity to improve the quality system;
(4) to meet regulatory requirements;
(5) to permit the listing of the audited organization's quality system in a register.

The reason for system auditing are to evaluate initially a supplier where there is a desire to establish a contractual relationship; to verify that an organization's own quality system continues to meet specified requirements and is being implemented; to verify that the supplier's quality system continues to meet specified requirements and is being implemented; and to evaluate an organization's own quality system against a quality system standard (ISO, 1990). An auditing process may be initiated for internal and/or external purposes, which is different from that of previous SQA methods, such as walkthrough, review and inspection, which are initiated purely for internal quality management.

The audit requirements and standards are set by internal and external sources. Most internal requirements and standards are dominated by the need to maintain product quality and meet customer needs. External sources of quality standards are usually identified by a quality agency, such as the International Standards Organization (ISO), the American Society for Quality Control (ASQC) and the Institute of Electrical and Electronics Engineers, Inc. (IEEE), as well as government agencies such as the Food and Drug Administration (FDA), Federal Aviation Administration (FAA), Department of Energy (DOE), Department of Defense (DOD), Environmental Protection Agency (EPA), etc. Some companies may wish to adopt software process management methods such as TQM, the capability maturity model (CMM) or ISO 9000 for quality control (Chou, 1996).

ISO auditing process. A stated *Guidelines for Auditing Quality Systems* describes the purpose and objectives of the auditing process (ISO/TC 176, 1990): "The 9000 series of international standards emphasizes the importance of quality audit as a key management tool for achieving the objectives set out in the organization's policy. The audit should be carried out in order to verify whether the various elements within a quality system are effective and suitable in achieving stated objectives. The quality system audit also provides objective evidence concerning the need for the reduction, elimination, and most importantly, prevention of noncomformities. The results of these audits can be used by management for improving the performance of the organization."

Companies seeking ISO 9000 certification should form an internal team to identify its own software quality assurance criteria, standards and procedures to implement its quality system, such as TQM or the CMM method (Chou,1996). Later, a qualified external audit registrar will be invited to inspect the company's quality system. This auditor should conduct the auditing process and check the actual practices and records to make sure that they are in compliance with the ISO quality system. If the results are appropriate, a compliance certificate is awarded. If any non-conformances are found during an audit, they must be corrected before certification can proceed. This certificate can be renewed every three to four years.

ISO-based auditing framework. The auditing framework establishes procedures and guidelines for performing the auditing process for an organization pursuing ISO-9000 certification. Chou (1996) proposes a seven-step auditing framework for pursuing ISO 9000 certification:

(1) Corporate management pursues ISO 9000 certification.
(2) An internal SQA team is formed to identify the quality system.
(3) Employees are trained on the quality system.
(4) The quality system is implemented.
(5) An external auditor is selected.
(6) The auditor performs the auditing process.
(7) The system is reviewed.

This framework proposes that corporations form an internal SQA team to identify the components, standards and objectives of a quality system. The corporate SQA team must develop a quality system training programme for employees. Management must support this quality system and enforce employees to follow such procedures. Software shops must implement the quality system after a training programme. All activities of the quality system implementation must be recorded and documented for future auditing usage. Later on, a selected external auditor should perform the auditing process to assess the conformance of the quality system. Auditing may be divided into four stages: preparation, performance, reporting and closure (Arter, 1994). Corporate executives should review the effectiveness of receiving certification. The results of certification guide corporate executives on setting up strategies for improving software quality and productivity.

TQM: Concept and applications

TQM has gained attention by academia and practitioners in recent years. Many organizations adopt TQM as their strategic policy. It is one of the surviving strategies in global competition. IS shops are also under increased pressure to deliver quality software to customers. Applying TQM to the software development process can control the software quality and productivity. This section discusses TQM concepts and methodologies, its application to the software development life cycle, and auditing in TQM.

TQM concepts and methods

TQM is a philosophy first adopted in Japan. It is an organizational management concept which concentrates on 'the voice of the customer'. There are many notable contributors to this discipline, including Joseph M. Juran, Philip B. Crosby, Kaoru Ishikawa, Armand V. Feigenbaum and W. Edwards Deming. Dr Deming is the prime scholar in this field.

Quality has been defined in a variety of ways (Flood, 1993):

(1) quality is a predictable degree of uniformity and dependability, at low cost and suited to the market (Deming, 1982);
(2) quality is fitness for use (Juran, 1980);
(3) quality is conformance to requirements (Crosby, 1984);
(4) quality is in its essence a way of managing the organization (Feigenbaum, 1986);
(5) quality is the totality of features and characteristics of a product, service or process, which bear on its ability to satisfy a given need, from the customer's viewpoint (British Standards Institution).

The key elements of TQM include the following: customer focus, obsession with quality, scientific approach, long-term commitment, teamwork, continual improvement of systems, education and training, freedom through control, unity of purpose, and employee involvement and empowerment. This list implies that total quality is an approach to do business that attempts to maximize the competitiveness of an organization through the continual improvement of the quality of its products, service, people, processes and environment (Goetsch & Davis, 1995).

The main principles of TQM can be identified as follows (Goetsch & Davis, 1995):

(1) there must be agreed requirements, for both internal and external customers;
(2) customers' requirements must be met first time, every time;
(3) quality improvement will reduce waste and total costs;
(4) there must be a focus on the prevention of problems, rather than an acceptance to cope in a fire-fighting manner;
(5) quality improvement can only result from planned management action;
(6) every job must add value;
(7) everybody must be involved, from all levels and across all functions;
(8) there must be an emphasis on measurement to help to assess and to meet requirements and objectives;
(9) a culture of continuous improvement must be established;
(10) an emphasis should be placed on promoting creativity.

Item (7) indicates that total quality means that everybody in an organization should be involved in the quality process, at all levels and across all functions, ensuring that quality is achieved according to the requirements in everything they do. TQM builds on the idea that an organization should be an interactive network of communication and control. This network provides full function of concurrent engineering, that is, several functional entities in one organization work together as a team to conduct decision-related activities (Chou, 1994b). Therefore, communication is a major contributor to a successful TQM programme.

SDLC in TQM

A SDLC comprises the complete activities of the IS process. It includes the phases of requirements analysis, systems design, systems development, testing, implementation and maintenance. A practical TQM programme should cover the complete practices of a SDLC.

The Deming cycle, also known as the P–D–C–A (plan–do–check–act) cycle in TQM, was developed to link the production of a product with consumer needs and to focus the resources of all departments in an organization (research, design, production, marketing) in a cooperative effort to meet those needs. This P–D–C–A cycle can be applied to the business process in organizations. First, the research department conducts consumer research and uses it in planning the product, i.e. the 'plan' stage. Second, producing the product is the 'do' stage. Third, checking the product to make sure it was produced in accordance with the plan is the 'check' stage'. Fourth, marketing the product is the 'act' stage. Finally the product has to be analysed to see how it is received in the market-place in terms of quality, cost and other criteria. These results will then be sent to the 'plan' stage for revising purposes, thus completing the Deming cycle.

The SDLC can be organized in accordance with the Deming cycle. The 'plan' stage accommodates the 'requirements analysis' phase of the SDLC. Software development requires resource coordination. The essence of human coordination is to facilitate communication about user requirements. Major software shops apply QFD to the software requirements analysis process. Software QFD focuses on improving the quality of the software development process by implementing quality improvement techniques to the requirements solicitation phase of the SDLC.

The 'do' stage accommodates the 'systems development' phase of the SDLC. Systems development requires system design and coding activities. Various systems requirements and specifications collected during the requirements analysis phase should be implemented at this phase. The action at this stage should proceed as planned. Records should be kept so that measurement can take place.

The 'check' stage accommodates the 'testing' phase of the SDLC. Testing activities should be planned and managed from the beginning of software development. The purpose of testing is to assess the system and to check a complete set of testing objectives. The success of software development needs to be measured against some accepted standards. Feedback from the customer is vital to the TQM process.

The 'act' stage accommodates the 'implementation and maintenance' phase of the SDLC. Software maintenance activities are usually reactive. Problems must be corrected and the process is often modified to improve the efficiency or product. Changes should be recorded and communicated to the customers. Any vital changes to the system should be forwarded to the system design team for future planning and development processes.

Auditing in TQM

An auditing process may be initiated for internal and/or external purposes. For example, an organization may initially evaluate a supplier for establishing a contractual relationship, and/or verify its own quality system to meet specified requirements and standards.

Companies seeking ISO 9000 certification should form an internal team to identify its own SQA criteria, standards and procedures. Companies may seek to implement a TQM-based quality system. Under such circumstances, the whole company should adopt the SQA methodology to analyze, design and develop software. All software production should follow these standards and all related activities should be documented. Later on, a qualified external audit registrar will be invited to inspect (audit) the company's quality system. If the results are appropriate, a compliance certificate is awarded. If any non-conformances are found during an audit, they must be corrected before certification can proceed. In the TQM paradigm, auditing deals with the last two steps in the P–D–C–A cycle, that is, measuring and improving activities (Arter, 1994).

Weakness of the existing auditing process

Chou's (1996) auditing framework indicates that the auditor's role is only during the last three of its seven stages. In the TQM paradigm, auditing deals with the last two steps in the P–D–C–A cycle (Arter, 1994). The traditional approach to auditing checks the conformance of a corporate quality system against its stated standards and objectives. The auditor, however, is responsible for the following duties (ISO, 1990):

(1) complying with the applicable audit requirements;
(2) communicating and clarifying audit requirements;
(3) planning and carrying out assigned responsibilities effectively and efficiently;
(4) documenting the observations;
(5) reporting the audit results;
(6) verifying the effectiveness of corrective actions taken as a result of the audit;
(7) retaining and safeguarding documents pertaining to the audit: submitting such documents as required, ensuring such documents remain confidential, and treating privileged information and discretion;
(8) cooperating with, and supporting, the lead auditor.

The traditional approach indicates that auditing work is reactive rather than proactive. The auditor can only perform both the 'check' and 'act' processes to ensure the conformance of stated objectives and standards. The quality of auditing depends on the clarity and completeness of the relevant documents, such as quality system plans, customer requirements and specification documents, feasibility reports, systems diagrams and illustrations, testing reports, etc. Any mistakes in these documents may mislead the auditing process and quality. One way to remedy this weakness in a TQM-based environment is to allow the auditor to take a proactive role in the SQA process. The auditor should be involved in this process at the beginning of the SDLC. He or she needs to attend the JAD session and understand the whole picture of the quality system and customer needs. This approach helps the auditor to perform a better job during later auditing activities.

Another weakness of the existing auditing process is the communication capability. Since auditing is based on documented information, any ambiguity found in documents may cost the auditor a lot of time to find the answer. Therefore, an early participation of the auditing process could fulfill the auditor's communication capability.

An integrated auditing process

A TQM approach must look at an organization as a whole. TQM delineates a whole system's view for quality management. Total quality means that everybody in an organization should be involved in quality, at all levels and across all functions. A TQM integrated auditing process should enable the following capabilities: interactive networks of communication and control, concurrent engineering, and a proactive and efficient auditing system in an organization.

Instead of performing auditing work during the latter session of the SDLC, the auditor should participate in the first phase of the SDLC. Along with customer, system analyst, customer support personnel and project manager, the auditor should participate at the requirements analysis phase. A JAD session may be used to extract the related information from customer during this phase.

Involvement of customer support personnel at the JAD session is highly appropriate, since they have the most contact with customers during the systems maintenance phase. The

presence of customer support personnel at the JAD session allows them to perceive the future system requirements and objectives. The auditor performs a similar function to that of customer support personnel. Early identification and understanding of the quality system's objectives, standards, requirements and specifications enables the auditor to perform better in the future. The systems analyst and project manager should facilitate the whole process of the JAD session. Interaction and communication among customers/users, the auditor, customer support personnel, systems analyst and project manager at an early stage, such as in the JAD session, could improve the quality of systems requirements analysis documents.

An integrated auditing process also aligns with TQM's disciplines: total involvement, full communication and concurrent engineering. A successful quality software system should meet customer requirements and needs. A full, communicated working environment could fulfill these needs. This quality management process ensures horizontal integration across networks, and vertical integration through hierarchies are achieved.

The change control process, as a part of software configuration management, contributes directly to software quality by formalizing requests for change, evaluating the nature of change and controlling the impact of change. Any requirement changes made by the customer during the SDLC should be communicated to the systems analyst, project manager, customer support personnel and the auditor. This approach ensures a fast requirement update on system design, coding and testing, system development and auditing processes.

A TQM integrated auditing process could enhance the effectiveness and efficiency of the auditing process. The auditor should be involved at the beginning and throughout the SDLC. Accurate and speedy communication guarantees the success of the auditing process.

Conclusions

As TQM becomes the mainstream of quality control, a suitable auditing framework is needed to guide the process of software quality assurance. The merits of TQM include concurrent engineering, total involvement, and a communication and coordination-oriented process. These merits can be utilized to advance the auditing framework in the SQA process. A TQM integrated auditing framework should ensure the capabilities of an interactive network of communications and control, concurrent engineering and a proactive auditing system in an organization. This auditing process can enhance corporate effectiveness and efficiency.

References

ARTER, D.R. (1994) *Quality Audits for Improved Performance* (Milwaukee, WI, ASQC Quality Press).

CHOU, D.C. (1994a) Software quality assurance in CASE tools: creating an AI enhanced CASE tool. In: *Information Systems Management*, Spring, pp. 56–61.

CHOU, D.C. (1994b) Decision network in multinational corporations. In: *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 183–187.

CHOU, D.C. (1996) Software quality auditing: its framework and tool design. In: *ACME Transactions 1996*, pp. 209–223.

CROSBY, P.B. (1984) *Quality without Tears* (New York, McGraw-Hill).

DEMING, E.W. (1982) *Quality, Productivity and Competitive Position* (Cambridge, MA, MIT).

FEIGENBAUM, A.V. (1986) *Total Quality Control* (New York, McGraw-Hill).

FLOOD, R.L. (1993) *Beyond TQM* (Chichester, John Wiley).

GOETSCH, D.L. & DAVIS, S. (1995) *Implementing Total Quality* (Englewood Cliffs, NJ, Prentice Hall).

ISO (1990) *Guidelines for Auditing Quality Systems Audits* (Geneva, International Standards Organization).

JOHNSON, J., SKOGLUND, R. & WISNIEWSKI, J. (1995) *Program Smarter, Not Harder* (New York, McGraw-Hill).

JURAN, J.M. (1980) *Quality Planning and Analysis*. (New York, McGraw-Hill).

PRESSMAN, R.S. (1992) *Software Engineering*, 3rd Edn (New York, McGraw-Hill).