Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 1998 Proceedings

Americas Conference on Information Systems (AMCIS)

December 1998

Critical Factors of Software Quality Management (SQM)

Padmal Vitharana
University of Wisconsin-Milwaukee

Mark Mone
University of Wisconsin-Milwaukee

Critical Factors of Software Quality Management (SQM)

Padmal Vitharana
Mark A. Mone
University of Wisconsin – Milwaukee

Abstract

The importance of managing quality in developing software systems is well documented. Existing literature presents attributes of good quality management practices in building software systems on time, within budget, and satisfying customer needs. However, most of the current findings on Software Quality Management (SQM) are case-based and company-specific, and may not be generalizable across all software developing organizations. Furthermore, the relationships examined in past research may be incomplete and therefore fail to identify differing dimensions critical for SQM.

Against this backdrop, our research provides a synthesis of existing literature by identifying six critical factors of SQM in software developing organizations. An instrument that can be used to measure critical factors of SQM is constructed. Validity and reliability of the instrument is established using a combination of exhaustive literature review, pre-pilot testing a preliminary version of the instrument among a group of academics, and pilot testing revised versions of the instrument among a group of IS professionals who are affected by software quality. An instrument such as the one proposed in this research could be used by senior IS managers to assess the effectiveness of SQM practices in developing quality software. Researchers on the other hand could use the instrument to better understand SQM practices, and build theories and models to manage software quality effectively.

Introduction

To survive in a globally competitive software industry, developers have to build systems on time and within budget. Recently, however, there has been an increasing emphasis on quality in developing software systems (Cortada, 1995). The quality of a software system is widely accepted as its conformance to customer requirements (Kan, Basili, and Shapiro, 1994). In order to develop software that meets customer expectations, quality efforts have to be managed effectively.

There have been some studies on how quality should be managed in developing software. Most current findings report SQM practices of successful companies by consultants, researchers, and managers. Kan et al. (1994) provide a high-level tutorial that discusses software quality in the context of Total Quality Management (TQM). TQM concepts, which have their roots in manufacturing, have recently been applied to software development. TQM philosophy emphasizes customer focus, top-management and employee participation on quality efforts, process improvements, and quality metrics. Dunn and Ullman (1994) and Arthur (1993) present further evidence of using TQM to improve software quality. The book edited by Schulmeyer and McManus (1992) brings together experience of a set of experts in applying TQM to software. Kaplan, Clark, and Tang (1995) present forty software quality innovations including management practices that helped IBM improve quality of its software products and processes. Horch (1996) provides a practical guide to SQM by identifying eight elements of a good software quality program. Jenner (1995) discusses SQM in the context of ISO 9001. ISO 9000 describes the International Standards Organization's (ISO) series of management systems standards. ISO 9001 titled "Quality systems - model for quality assurance in design, development, production, installation, and servicing," is the one frequently applied to software.

Quality Management Research

Early quality management research focused mainly on manufacturing organizations. Deming, Crosby, and Juran are recognized as the pioneers of quality management research. Deming (1986) prescribes 14 principles for effective quality management in an organization. Crosby (1979) presents 14-steps toward establishing an effective quality program. Juran (1986) identifies the "Quality Trilogy," quality planning, quality improvement, and quality control. Some of the general quality management practices endorsed by these researchers can also be extended to software development. For example, Deming's prescription of strong management commitment to quality and company-wide training and education in quality is equally applicable to software. Crosby also prescribes management commitment and training as well as establishment of a quality measurement system that can be easily adopted by software developing organizations to better manage their software quality programs. Similarly, Juran's prescriptions such as quality planning and establishment of a formal quality policy are likewise pertinent to effective SQM.

Current Research on SQM

Although there is some literature on SQM, it is mostly case-based and company-specific, and may not be generalizable across all software developing organizations. So far, there has been no effort to methodically collect and synthesize various critical factors identified by different authors. Moreover, measures of effective SQM practices have not been proposed. Although most software developing firms collect quality performance measures such as number of defects, to our knowledge, there is no comprehensive instrument to measure critical factors of SQM. For example, top-management commitment to quality

has been identified as one of the factors in SQM (see Table 1 for a list of citations), but no operational measures are currently available for this critical factor. In addition to the problem of generalizability of existing studies, the relationships examined in these studies are incomplete, and therefore fail to identify differing dimensions critical to SQM. This research synthesizes existing literature by identifying six critical factors of SQM in developing software. Using these six factors, an instrument to measure critical factors of SQM is constructed.

There is a profusion of literature in general quality management, especially that of manufacturing, from which SQM researchers have drawn. Early quality management literature also lacked efforts in systematic collection and organization of critical factors of general quality management. In realizing this gap, Saraph, Benson, and Schroeder (1989) organized and synthesized quality literature to identify eight critical factors of quality management in manufacturing and service firms, and presented an instrument for measuring these critical factors. In operationalizing the instrument, they considered 73 quality managers and general managers of various divisions in 10 service companies, of which none can be considered software-developing organizations. Although considered part of the service industry, software differs from other services such that it entails the development of a software product, and that quality issues have to be addressed from early requirements analysis to final delivery, installation, and maintenance of that software product. The study undertaken in this research attempts to collect and synthesize critical factors of SQM similar to that of Saraph et al. (1989) who collected and synthesized critical factors of quality management in manufacturing and service firms. At the same time, this research addresses quality management in the specific context of software development in contrast to that of Saraph et al. (1989) who addressed quality management in a more general context.

An instrument such as the one developed in this research would be useful for both practitioners and researchers in identifying critical factors in SQM. Practitioners could use the instrument to identify strengths and weaknesses of their SQM practices. Once weaknesses are identified, managers will be able to manipulate relevant factors in quality management to improve software quality. Furthermore, the instrument can be also used to compare and contrast the effectiveness of SQM practices across IS departments of different software developing organizations. Researchers on the other hand could use the instrument to better understand industry practices in managing software quality and build theories and models that relate the critical factors of SQM to the quality of the software product being built. For example, with a thorough understanding of critical factors of SQM, researchers will be able to build theories and models to map management practices with software quality criteria (e.g., correctness, reliability, etc.).

Critical Factors of SQM

To identify critical factors in SQM, an exhaustive literature review was conducted. This review of SQM literature focused on those articles, books, and studies that were obtained from a computer-based search on ABI/INFORM and CARL library databases. Consequently, a comprehensive set of six critical factors of SQM that consists of management commitment, education and training, customer focus, process management, quality metrics, and employee responsibility is identified. These critical SQM determinants are synthesized in Table 1 and are briefly discussed below.

- *Management commitment* to quality in terms of resource allocation, staffing, personnel evaluations and rewards, and providing the necessary leadership to create an overall quality culture.
- *Education and training* for both managers and developers as a prerequisite to building quality software.
- *Customer focus* in achieving total customer satisfaction through studying customers' wants and needs, gathering customer requirements, and measuring customer satisfaction.
- *Process management* to increase visibility into the software process, and therefore enhance predictability of software quality.
- *Software metrics* to identify various attributes that need to be measured and how to measure them.
- *Employee responsibility* for developing quality software.

Current Status

The literature review resulted in an initial survey with 76 items of SQM. After pre-pilot testing the instrument among 10 management information systems and computer science faculty members of a major metropolitan university in the Midwest, we have revised, and as a result reduced the survey to 57 items. We are currently in the process of pilot testing the survey among a group of IS professionals in the Midwest region. Using this data, we will validate the instrument.

The independent variable in this research is a set of SQM practices while the dependent variable is software quality. It is our hope that the proposed instrument could accurately assess the effectiveness of SQM practices in developing quality software.

References

Arthur, L. J. *Improving Software Quality: An Insider's Guide to TQM*, Wiley, New York, NY, 1993.

Billings, C., Clifton, J., Kolkhorst, B., Lee, E., and Wingert, W. B. "Journey to a Mature Software Process," *IBM Systems Journal*, (33:1), 1994, pp. 46-61.

Cortada, J. W. *TQM for Information Systems Management: Quality Practices for Continuous Improvement*, McGraw-Hill, New York, NY, 1995.

Crosby, P. B. *Quality is Free*, New American Library, New York, NY, 1979.

Deming, W. E. *Out of the Crisis*, MIT Center for Advanced Engineering, Cambridge, MA, 1986.

Dunn, R. H. and Ullman, R. S. *TQM for Computer Software* (2nd ed.), McGraw-Hill, New York, NY, 1994.

Horch, J. W. *Practical Guide to Software Quality Management*, Artech House Publishers, Boston, MA, 1994.

Jenner, M. G. *Software Quality Management and ISO 9001: How to Make Them Work for You*, Wiley, New York, NY, 1995.

Juran, J. M. "The Quality Trilogy," *Quality Progress*, (9:8), 1986, pp. 19-24.
Kaplan, C., Clark, R., and Tang, V. *Secrets of Software Quality: 40 Innovations from IBM*, McGraw-Hill, New York, NY, 1995.
Kan, S. H., Basili, V. R., and Shapiro, L. N. "Software Quality: An Overview from the Perspective of Total Quality Management," *IBM Systems Journal*, (33:1), 1994, pp. 4-19.
Kan, S. H. *Metrics and Models in Software Quality Engineering*, Addison-Wesley, Reading, MA 1995.
Saraph, J. V., Benson, P. G., and Schroeder, R. G. "An Instrument for Measuring Critical Factors of Quality Management," *Decision Sciences*, (20:4), 1989, pp. 810-829.
Schulmeyer, G. G. and McManus, J. I. (Eds.). *Total Quality Management for Software*, Van Nostrand Reinhold, New York, NY, 1992.
Yeh, H. *Software Process Quality*, McGraw-Hill, New York, NY, 1993.

Table 1. Critical Factors of Software Quality Management and Corresponding References

Critical Factors of SQM	Corresponding References
Management Commitment	Schulmeyer and McManus (1992); Arthur (1993); Yeh (1993); Kan, Basili, and Shapiro (1994); Cortada (1995); Dunn and Ullman (1995); Jenner (1995); Kaplan, Clark, and Tang (1995); Horch (1996)
Education and Training	Schulmeyer and McManus (1992); Arthur (1993); Cortada (1995); Dunn and Ullman (1995); Jenner (1995); Kaplan, Clark, and Tang (1995); Horch (1996)
Customer Focus	Schulmeyer and McManus (1992); Arthur (1993); Kan, Basili, and Shapiro (1994); Cortada (1995); Dunn and Ullman (1995); Jenner (1995); Kaplan, Clark, and Tang (1995)
Process Management	Schulmeyer and McManus (1992); Arthur (1993); Yeh (1993); Billings, Clifton, Kolkhorst, Lee, and Wingert (1994); Kan, Basili, and Shapiro (1994); Cortada (1995); Dunn and Ullman (1995); Jenner (1995); Kaplan, Clark, and Tang (1995); Horch (1996)
Quality Metrics	Schulmeyer and McManus (1992); Arthur (1993); Kan, Basili, and Shapiro (1994); Jenner (1995); Kan (1995); Horch (1996)
Employee Attributes	Schulmeyer and McManus (1992); Arthur (1993); Kan, Basili, and Shapiro (1994); Cortada (1995); Jenner (1995); Kaplan, Clark, and Tang (1995); Horch (1996)