

Chapter 12

Software Quality Management: Measurement and Research Directions

Padmal Vitharana
Syracuse University, USA

Mark A. Mone
University Wisconsin - Milwaukee, USA

ABSTRACT

While the value of the Capability Maturity Model (CMM), ISO 9000, and Total Quality Management (TQM) concepts in managing software quality has been widely acknowledged, shortcomings of these approaches have also been recognized. The current research synthesizes existing literature bases in CMM, ISO 9000, TQM, among others, to identify six critical factors of Software Quality Management (SQM) and then develops an instrument that can be used to measure critical factors of SQM. Validity and reliability are established by reviewing existing literature, testing a preliminary version of the instrument among a group of researchers and industry experts, and empirically testing a revised version of the instrument among a group of IS professionals. The authors conclude by addressing quality management research issues in the emerging open source software (OSS) paradigm.

INTRODUCTION

Recently, there has been an increasing emphasis on quality in developing software (Duggan, 2004; Harter et al. 2003; Jureta et al. 2009; Prajogo & Sohal, 2006; Tarvo, 2009). The quality of a software system is widely accepted as its conformance to customer requirements (Kan et al., 1994). The interest in quality is heightened as more system failures are attributed to issues in software quality

that often lead to higher maintenance costs, longer cycle times, customer dissatisfaction, lower profits, and loss of market share (Arthur, 1993; Gopal et al., 2002; Kan et al., 1994; Tarvo, 2009). Although the importance of quality is acknowledged, managing quality efforts remains a major challenge in software development. In this context, software quality management refers to the process of carrying out key management practices necessary for achieving software quality.

The problems associated with inadequate software quality belie the amount of research on

DOI: 10.4018/978-1-60566-962-5.ch012

how software quality should be managed. Such normative work typically reports software quality management practices of successful companies by consultants, researchers, and managers. For example, Kan et al. (1994) discuss software quality in the context of Total Quality Management (TQM). Although the TQM philosophy in general emphasizes continuous improvement in quality, various TQM advocates prescribe a diverse array of techniques for quality management. Several authors present evidence of using TQM to improve software quality (e.g., Arthur 1993; Dunn & Ullman, 1994; Issac et al. 2006; Prajogo & Sohal, 2006; Manz & Stewart 1997; Subramanian et al. 2007; Victor et al., 2000). For instance, Ravichandran and Rai (2000) apply TQM principles to information systems development and derive a set of scales for assessing quality constructs. Others have acknowledged differences between soft (behavioral) TQM factors such as employee commitment and hard (technical) TQM such as statistical testing and their implications for quality performance (e.g., Rahman et al., 2005). Still others have examined the mediating role of TQM on the relationship between firm strategy and firm performance (Prajogo & Sohal, 2006). TQM, however, represents only one stream of research applicable to the management of software quality.

Software quality management is also discussed in the context of ISO 9001, ISO 9000-3, and SPICE (Jenner, 1995; Jung, 2005; Kehoe & Jarvis, 1996; McManus & Wood-Harper, 2007; Pino et al., 2008; Yoo et al., 2006). By achieving ISO certification, an organization is able to conduct business with customers or vendors who require that their partners adhere to accepted quality standards. However, because of possible disruption of normal operations and long duration of the auditing process, the costs involved in ISO certification could be considerable to some organizations. Moreover, because ISO 9001 only defines minimum qualifications a firm needs to achieve for certification (Bamford et al., 1993), it lacks

support for continuous improvement in software quality (Coallier, 1994). In addition, ISO 9001's high level of abstraction has caused auditors to interpret it in different ways (Paulk, 1995).

The Capability Maturity Model (CMM¹) developed by Software Engineering Institute (SEI) details a well-defined approach to software process improvement (Harter et al., 2000; Manzoni & Price, 2003; McManus & Wood-Harper 2007). While ISO 9000 highlights a more inter-organizational approach (e.g., vendor relationships) to managing quality, CMM takes a more intra-organizational strategy to quality management. Although CMM has been used in the industry as a means to assessing software process maturity, there is some criticism of its 5-level maturity model. For example, Saiedian and Kuzara (1995) claim that CMM is not an exhaustive model in that it does not address several software management and engineering practices crucial for project success and that because of CMM's roots in the governmental and defense-oriented software arena, its assumptions may not necessarily hold true in the commercial sector. Furthermore, Stelzer et al. (1997) highlight that CMM lacks a solid understanding and an explicit model for explaining the causes and effects of software quality management. More recently, Niazi et al. (2005a) observe that because of the complexity of CMM, little attention has been paid to its implementation, resulting in limited success in many software process improvement efforts. Notwithstanding their shortcomings, ISO 9000 and CMM have helped several organizations improve their ability to develop quality software (Harter et al., 2000). More recently, the emergence of the Capability Maturity Model Integration (CMMI) and ISO's SPICE typify the continued importance of the software process (Chrissis et al., 2003; Jung, 2005; Niazi, Wilson, & Zowghi, 2005b; Yoo et al., 2006). Specifically, CMMI provides guidance for improving the organization's processes, and managing the development, acquisition and maintenance of products and services. On the other

21 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/software-quality-management/39244

Related Content

The Impact of IT Personnel Skills on IS Infrastructure and Competitive IS

Terry Anthony Byrd, Bruce R. Lewis and Douglas E. Turner (2004). *Information Resources Management Journal* (pp. 38-62).

www.irma-international.org/article/impact-personnel-skills-infrastructure-competitive/1255

Organizational Assimilation Capacity and IT Business Value

Vincenzo Morabito and Gianluigi Viscusi (2009). *Encyclopedia of Information Science and Technology, Second Edition* (pp. 2929-2933).

www.irma-international.org/chapter/organizational-assimilation-capacity-business-value/14006

Increasing Diversity in Cybersecurity: Strategies for Engaging High School Students of Color

Jude Dominique Joseph (2025). *International Journal of Information Systems and Social Change* (pp. 1-19).

www.irma-international.org/article/increasing-diversity-in-cybersecurity/381680

Improvement in Operational Efficiency Due to ERP Systems Implementation: Truth or Myth?

Vijay K. Vemuri and Shailendra C. Palvia (2008). *Innovative Technologies for Information Resources Management* (pp. 43-61).

www.irma-international.org/chapter/improvement-operational-efficiency-due-erp/23845

A Suggested Integrated Approach for the Evaluation of Remote Intranet Usability: A Case Study of Remote Workers During a Pandemic

Ghada Refaat Mohamed El Said (2022). *Information Resources Management Journal* (pp. 1-22).

www.irma-international.org/article/a-suggested-integrated-approach-for-the-evaluation-of-remote-intranet-usability/287905