Chapter 2.6
Measuring Critical Factors of Software Quality Management:
Development and Validation of an Instrument

Padmal Vitharana
Syracuse University, USA

Mark A. Mone
University Wisconsin, USA

ABSTRACT

Literature presents attributes of effective quality management in building software systems. The value of the capability maturity model (CMM), ISO 9000, and total quality management (TQM) concepts in managing software quality has been widely acknowledged. However, shortcomings of these approaches have also been recognized. For instance, scholars have acknowledged CMM implementation difficulties because of its inherent complexity. This research synthesizes existing literature base 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 extant 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 instrument is relatively easy to implement and requires minimal resources. Implications for research and practice are discussed.

INTRODUCTION

Recently, there has been an increasing emphasis on quality in developing software (Duggan, 2004; Haag, Raja, & Schkade, 1996; Harter & Slaughter, 2003; Prajogo & Sohal, 2006). The quality of a software system is widely accepted as its conformance to customer requirements (Kan, Basili, & Shapiro, 1994). The interest in quality is heightened as more system failures are attrib-

uted 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, Krishnan, Mukhopadhyay, & Goldenson, 2002; Kan, Basili, & Shapiro, 1994). For instance, a software problem with one of the switches left AT&T customers nationwide without service for 26 hours, costing the company about $1 billion and 6.25 cents per share loss (New York Times, 1998). In a more recent report, AT&T claims that it spends roughly $1 million a month to patch defective software built by others (Wall Street Journal, 1998). 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 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; Prajogo & Sohal, 2006; Manz & Stewart 1997; Victor, Boynton, & Stephens-Jahng, 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 & Bullock, 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 and ISO 9000-3 certification, and more recently, SPICE (Jenner, 1995; Kehoe & Jarvis, 1996; Jung, 2005; Yoo, Yoon, Lee, & Lee, 2006). By achieving ISO certification, an organization is able to conduct business with those (e.g., 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 & Deibler, 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 (CMM[1]) developed by Software Engineering Institute (SEI) details a well-defined approach to software process improvement (Harter, Krishnan, & Slaughter, 2000; Manzoni & Price, 2003). 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

Related Content

Selecting Indispensable Edge Patterns With Adaptive Sampling and Double Local Analysis for Data Description
Huina Liand Yuan Ping (2024). *Journal of Cases on Information Technology (pp. 1-26).*
www.irma-international.org/article/selecting-indispensable-edge-patterns-with-adaptive-sampling-and-double-local-analysis-for-data-description/335945

The Improved SIFT Algorithm Based on Rectangular Operator and Its Parallel Implementation
Weiwei Yang, Jing Yangand Haifeng Song (2019). *Journal of Information Technology Research (pp. 1-17).*
www.irma-international.org/article/the-improved-sift-algorithm-based-on-rectangular-operator-and-its-parallel-implementation/216395

Towards Modelling the Impact of Security Policy on Compliance
Winfred Yaokumah, Steven Brownand Alex Ansah Dawson (2016). *Journal of Information Technology Research (pp. 1-16).*
www.irma-international.org/article/towards-modelling-the-impact-of-security-policy-on-compliance/160154

Reality of Use and Nature of Change in Small Business: A Case Study in Inefficient Compromise
Wita Wojtkowskiand J. Craig Hardesty (2001). *Annals of Cases on Information Technology: Applications and Management in Organizations (pp. 217-225).*
www.irma-international.org/chapter/reality-use-nature-change-small/44617

An Analysis of Intranet Infusion Levels
Lauren B. Eder, Bay Arinze, Marvin E. Darterand Donald E. Wise (2000). *Information Resources Management Journal (pp. 14-22).*
www.irma-international.org/article/analysis-intranet-infusion-levels/1212