Chapter XV

Reducing IS Maintenance by Improving the Quality of IS Development Processes and Practices

Evan W. Duggan, University of Alabama, USA

ABSTRACT

The correlation between the quality of information systems (IS) development practices, the products they shape, and systems maintainability has been well established. Several organizations have expended large amounts of money on ineffective software that have attracted high maintenance activities, which consume a disproportionate share of IS development resources. The IS quandary is how to reverse this trend, and free up resources for more productive organizational endeavors. This chapter presents a review of a variety of IS quality-oriented approaches (based on research findings and dispassionate practitioner accounts) and an indication of their advantages and weaknesses, and the development environments and contexts for which they are most suited. These techniques are available to software developers to be used individually or in synergistic combinations to confront the prevalent problem of poor software quality and reduce its unfavorable impact on software maintenance.

INTRODUCTION

Organizations invest in information systems (IS) to apply information technology (IT) in support of their business processes against the backdrop of the extremely short half-life of IT and typically volatile business environments. The goal of IS maintenance, therefore, is to preserve some measure of stability (during the longest period of a system’s life cycle) by keeping the application in congruence with the business
operations it enables (Floyd, 1987). All systems (including IS) may degrade because of atrophy—the propensity of entities to deteriorate for lack of sustenance, or entropy—impairment due to natural decay. IS stability is directed at preventing the former and delaying the latter to leverage an organization’s IS investment over a longer period.

Maintenance activities have progressively consumed a larger share of systems development budgets (Banker et al., 1998; Pressman, 2000; Swanson & Beath, 1989): approximately 40 percent in the 1970s; 60-70 percent in the 1980s; and up to 80 percent in the 1990s (Hoffer et al., 2002). But beyond this high resource allocation, organizations bear at least two other often overlooked systems maintenance costs: the direct cost of systems downtime, estimated by the Standish Group at $10,000 per minute for mission-critical applications (Schneider, 1999) and the opportunity cost of resources devoted to software maintenance, which could be invested in more productive organizational pursuits (Banker & Slaughter, 1997; Keil, 1995).

Consequently, several researchers (Banker et al., 1991; Dishaw & Strong, 1998; Prechelt et al., 2001; Ramanujan et al., 2000) have focused on understanding this phenomenon of escalating IS maintenance and have prescribed improvements to the maintenance process. They have analyzed the effects of factors such as software complexity, size, age, documentation, user population, program structure, capability of the maintenance group, and the stability of the business environment on maintenance practices.

Others, such as Banker et al. (1993), Banker and Slaughter (1997) and Boehm (1981) have concentrated on maintenance economies and cost-reducing mechanisms. Yet a third group of researchers (Banker et al., 1991, Dekleva, 1992; Gibson & Senn, 1989; Ghods & Nelson, 1998, Swanson & Dans, 2000; Vessey & Weber, 1984; Zagal et al., 2002) have explored the downstream effect of the quality of IS development practices and delivery processes on system maintainability and the maintenance experience. System maintainability has been defined as the ease and speed with which systems can be understood and modified to correct errors, adapt to changing business environments, and improve performance (Bandi, Vaishnavi & Turk, 2003; Oman & Hagemeister, 1994; Pressman, 2001; Zhuo et al., 1993). The discussion in this chapter extends the latter theme and describes a set of quality-focused, IS delivery approaches that could induce maintainability as a de facto software development goal.

There is empirical confirmation of the proposition that mature systems delivery processes and effective methods produce high quality systems (Harter et al., 1998; Kalifa & Verner, 2000; Ravichandran & Rai, 1996). Similarly, there is enormous support among researchers for a multivariate attack on the problem of poor software quality with deliberate and focused combinations of available methods, matched specifically to the software application context (Broady et al., 1994; Fitzgerald, 1998; Hocking, 1998; Jaakkola & Drake, 1991; Korac-Bosvert & Kouzmin, 1995; Lee & Xia, 2002; Meso & Medo, 2000; Saberwal & Robey, 1995; Vessey & Glass, 1998; Vlasblom et al., 1995). However, the allure has not necessarily caught on in practitioner communities, where several false alarms of silver bullets have left a great deal of ambivalence about the legitimacy of some approaches.

In this chapter, I synthesize elements of these streams of research to make three important contributions to the theme of managing software maintenance in organizations. First, I provide an extensive review of the literature on IS quality, maintenance practices, and information systems development approaches with implications for
Related Content

Product Configuration in ETO Companies
www.igi-global.com/chapter/product-configuration-eto-companies/26119?camid=4v1a

Model-Driven Reverse Engineering of Open Source Systems
Ricardo Perez-Castillo and Mario Piattini (2014). Information Systems and Technology for Organizational Agility, Intelligence, and Resilience (pp. 139-160).
www.igi-global.com/chapter/model-driven-reverse-engineering-of-open-source-systems/107106?camid=4v1a

Adoption of a New Online Travel Management System for FED-AK
www.igi-global.com/chapter/adoption-new-online-travel-management/69107?camid=4v1a
Business Risk Analysis: Obsolescence Management in Requirements Engineering
www.igi-global.com/chapter/business-risk-analysis/63247?camid=4v1a