Available online at www.sciencedirect.com
SCIENCE DIRECT®
Information & Management 41 (2004) 279–288
INFORMATION & MANAGEMENT
www.elsevier.com/locate/dsw

An exploration of the relationship between software development process maturity and project performance☆

James J. Jiang[a,*], Gary Klein[b], Hsin-Ginn Hwang[c], Jack Huang[c], Shin-Yuan Hung[c]

[a]*Department of Management Information Systems, University of Central Florida, Orlando, FL 32816-1400, USA*
[b]*University of Colorado, Colorado Springs, USA*
[c]*National Chung Cheng University, Jai-Yi Taiwan*

Accepted 30 June 2003

Abstract

Software projects have a high rate of failure. Organizations have tried to reduce the rate through methodological approaches but with little perceived success. A model of software development maturity (the capability maturity model (CMM)) describes managerial processes that can be used to attack software development difficulties from the managerial control perspective at five maturity levels. Our study examined performance of projects in relation to the activities at these various levels of maturity. A survey of software engineers indicated that the activities associated with the managerial control of development related positively to project performance measures. However, not each level of maturity demonstrated observable benefits, indicating that greater caution is needed in the planning and implementation of the activities.

Keywords: Software development; Capability maturity model; Project performance

Software projects continue to grow more critical to the organizations that employ them. Still, software failures are frequent. According to Standish Group International [38], about 15% of all software development projects never deliver a final product. Software project problems cost US companies and government agencies an estimated US$ 145 billion annually. In large software development projects, more than 80% are excessively late and over budget [13,26]. One approach to combat the failure rate has been technical, with organizations introducing new design methods [4,25]. However, after significant resources are poured into software development methods, such as CASE and Rapid Application Development, projects are often still considered ''run-aways'' [2,5,28,39,40].

The managerial side of the software development project, meanwhile, is often conducted without adequate planning, with poor understanding of the overall development process, and a lack of a well-established management framework [36,37]. A formal addition to management practice was developed in the 1990s to help organizations along an evolutionary path to a more disciplined development process. The Software Engineering Institute (SEI), for the US Department of Defense (DoD), recommended a number of key software process improvement (SPI) areas. Later, these activities were formed into an evaluative framework called the capability maturity model (CMM) [33,34].

☆ For the Research Section of Information & Management.

* Corresponding author. Tel.: +1-407-823-4864; fax: +1-407-823-2389.
E-mail address: jjiang@bus.ucf.edu (J.J. Jiang).

0378-7206/$ – see front matter
doi:10.1016/S0378-7206(03)00052-1

This model requires a considerable amount of time and effort to implement and often needs a major shift in culture and attitude [7,19]. One study found that the median time for an organization to move up one level of the five-level CMM is between 21 and 37 months [16]. Over three-quarters of the organizations reported that implementing any key SPI activity took longer than expected. In fact, resources expended amount to billions of dollars every year in the US alone. In addition to these, an organization's culture can be adversely impacted by adding to its rigid bureaucracy and reducing creativity and freedom on the part of the developers [21]. Thus, given the costs and potential disadvantages of implementing the CMM, benefits must be evident to justify its continued use.

The CMM has already achieved wide interest and acceptance in the software industry. It has spread far beyond its origins and is now used by thousands of major organizations worldwide [12]. But is the CMM an appropriate model for guiding improvements in the software development process? Do the suggested SPI activities really apply to a variety of organizations? To our surprise, no empirical evidence examining the effects of the various suggested SPI activities on project performance can be found in the information system (IS) literature. Therefore, the focus of this study is on the relationship between SPI activities and software project performance. Specifically:

1. Is there a relationship between the implementation of the CMM's SPI activities and an organization's software project performance?
2. Are certain SPI activities more likely to influence the final project outcomes than other suggested SPI activities?
3. Which dimensions of project outcomes are more likely to be influenced by which SPI activities?

1. The software capability maturity model and software project performance

The CMM was originally developed by the Software Engineering Institute and it has been enhanced since then. It was primarily based on the experiences and extensive feedback of software practitioners and designed to assist the US Department of Defense in software acquisition. In September 1987, the SEI released a brief description of the process-maturity framework. It described an evolutionary software development process improvement path from an ad hoc, immature process to a mature, disciplined process. The CMM maps organization's software project

Table 1
Key process activities for CMM

Maturity levels	Characteristics	Key process activities
Level V: Optimizing	Process optimization	Process change management Technology change management Defect prevention
Level IV: Managed	Measuring quality of development process and its product	Software quality management Quantitative process management
Level III: Defined	Processes engineering	Peer reviews Intergroup coordination Software product engineering Integrated software management Training program Organization process definition Organization process focus
Level II: Repeatable	Basic project management	Software configuration management Software quality assurance Software subcontract management Software project tracking and oversight Software project planning Requirements management
Level I: Initial	Chaotic, few if any process	None