Available online at www.sciencedirect.com
SCIENCE DIRECT®
Information & Management 43 (2006) 297–307
INFORMATION & MANAGEMENT
www.elsevier.com/locate/dsw

Selection priority of process areas based on CMMI continuous representation

Sun-Jen Huang *, Wen-Ming Han

Department of Information Management, National Taiwan University of Science and Technology, 43, Sec. 4, Keelung Road, Taipei 106, Taiwan

Received 5 April 2004; received in revised form 2 May 2005; accepted 23 August 2005
Available online 7 October 2005

Abstract

An essential decision that must be made by software organizations that adopt the continuous representation of the capability maturity model integration (CMMI) for software process improvement concerns a suitable path that best meets their business objectives and mitigates the organization's risk. However, the CMMI models released by the Software Engineering Institute do not give their adopters any guidance on how to make such a decision. Thus, managers often make subjective selections of the areas in which to implement process improvement. Our study presents a decision support model that assists managers in determining the priorities of the CMMI process areas based on the characteristics of the is being developed. The proposed model was validated by using the ISBSG repository, and an example is presented to demonstrate the application of the model. Given the fact that hardly any research has yet been done on how to select the CMMI process areas to initialize process improvement, this study provides a starting point for the community in considering this important issue.

Keywords: Capability maturity model integration; Software process improvement; Critical success process areas; Project management; Information system characteristics

1. Introduction

A well-defined software development process can improve software quality and reduce the risks of software development projects. The Software Engineering Institute (SEI) at Carnegie Mellon University firstly released the capability maturity model for software (SW-CMM) Version 1.0 in 1991 to evaluate the maturity of software development of US Department of Defense contractors and provide a roadmap for software process improvement (SPI). The SEI solicited feedback from its adopters and then released SW-CMM Version 1.1 in 1993 [23]. Thereafter, SW-CMM Version 1.1 became a widely accepted benchmark for software organizations in both the initiation of their SPI efforts and evaluation of the maturity of their software development processes [1,8,16]. According to the SEI maturity profile in August 2004 [25], software organizations spent an average of 2 years to raise the level of their process maturity if they choose the staged representation of a CMMI model.

Since SEI released SW-CMM Version 1.1, it has been applied to different areas; hence, many capability maturity models have been announced. These included the software acquisition CMM (SA-CMM) [6], system engineering CMM (SE-CMM) [11], integrated product development CMM (IPD-CMM) [26] and people CMM (P-CMM) [2]. These models were developed by various organizations, and so they had overlapping scopes of

* Corresponding author. Tel.: +886 2 2737 6779; fax: +886 2 2737 6777.
E-mail address: huangsj@mail.ntust.edu.tw (S.-J. Huang).

0378-7206/$ – see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.im.2005.08.003

applications and lacked consistency in architecture, terminology, and assessment methodology. These problems have increased the cost and time to implement multiple model-based process improvement. Therefore, SEI released capability maturity model integration (CMMI) in 2001 to integrate existing capability maturity models. Its primary advantages are: in the elimination of the inconsistencies and duplication and thus streamline the enterprise-wide process improvement, in the reduction of the cost and time associated with model-based process improvement, and thus in the increase of return on investment of organizational SPI efforts [3,4].

To accommodate different process-improvement needs for software organizations, the CMMI product team provided them with two choices (staged or continuous representation) to increase the maturity of their processes. The CMM staged representation provided a framework for organizing the evolutionary steps into five levels of maturity (initial, managed, defined, quantitatively managed, and optimizing). These are ordinal scales for measuring the maturity of an organization's software process that can also be used for its internal process improvement. SEI then added the continuous representation to the CMMI for providing flexibility to enable software organizations to choose their improvement paths. The continuous representation allows comparisons of a specific process area across software organizations. This representation therefore allowed process improvement to be compared with that of the ISO/IEC 15504 standard.

When a software organization adopts the continuous representation of a CMMI model, the foremost decision of the project manager is to select the order of implementing the process areas that best meet the organizational business objective [4]. An unsuitable path can negatively affect the results of the activities, thus reducing enthusiasm for software process improvement. However, the CMMI models currently released do not provide any guidance to software organizations adopting the CMMI continuous representation on the priorities of developing the process areas.

Recent reviews of empirical studies of software project management have indicated that a software framework must be configured according to the specific characteristics of the project [14,22,24,27]. Also an improvement path must be configured according to the specific software development environment. The characteristics of the software being developed cru-

Table 1
Process areas in CMMI-SW/SE staged representation

Maturity level (ML)	Focus	Process area
ML 5: Optimizing	Continuous process improvement	Organization innovation and deployment (OID) Causal analysis and resolution (CAR)
ML 4: Quantitatively managed	Quantitative management	Organization process performance (OPP) Quantitative project management (QPM)
ML 3: Defined	Process standardization	Requirements development (RD) Technical solution (TS) Product integrated (PI) Verification (VER) Validation (VAL) Organizational process focus (OPF) Organizational process definition (OPD) Organizational training (OT) Integrated project management (IPM) Risk management (RSKM) Decision analysis and resolution (DAR)
ML 2: Managed	Basic project management	Requirements management (REQM) Project planning (PP) Project monitoring and control (PMC) Supplier agreement management (SAM) Measurement and analysis (MA) Process and product quality assurance (PPQA) Configuration management (CM)
ML 1: Initial	ad hoc process	None of process areas