An Analysis of Excluded IS Processes in the Capability Maturity Model and Their Potential Impact

Russell L. Purvis
University of Central Florida

Jose Santiago
University of Central Florida

V. Sambamurthy
Florida State University

Process improvement efforts are becoming pervasive within IS organizations as they attempt to meet the growing challenges of today’s complex and dynamic environment. A popular framework often used in these efforts is the Capability Maturity Model (CMM) developed by the Software Engineering Institute. The model focuses specifically on software process improvement, neglecting the other functions of IS. This paper assesses which IS functions are excluded by the CMM by comparing it to a earlier, more comprehensive model, the Information Systems Management Architecture developed by IBM. The comparison of models is followed by a discussion of the potential ramifications of a model with such a narrow focus.

As an ever larger number of businesses embrace the principles of quality management and process improvement, IS organizations are feeling growing pressure to follow suit and implement similar types of programs to improve systems development and operations services. IS organizations embarking in these programs often follow a standardized, step-by-step methodology to improve their processes.

The prevailing standardized process improvement methodology for IS organizations is Carnegie Mellon Software Engineering Institute’s Software Process Maturity Framework. The original intent of the Software Process Maturity Framework was to provide a methodology for improving the software development processes and assess the software engineering capabilities of contractors (Humphrey, 1989). This framework has since undergone a number of revisions and become what is now known as the Capability Maturity Model (CMM), a model which applies the process management concepts of TQM to the software engineering processes (Paulk, 1995).

The CMM is specific to the management and development of software products and excludes the processes of the remaining IS organization. Paulk (1995) explains, “The CMM is not a silver bullet and does not address all of the issues that are important for successful projects (pg. 13).” He further explains, “Since the CMM is focused on software issues, there are a number of other issues that should be considered as part of an overall improvement program, but which are only touched on in the CMM (pg. 89).” The basic research question for this article is: what are the consequences of using an incomplete measurement tool when assessing quality improvement programs for an IS organization? More specifically:
RQ1: What IS processes outside of the management and development of software products are excluded from the CMM process improvement framework?

RQ2: What are the potential ramifications of focusing process improvement programs on systems development and not considering all IS processes?

To better understand these issues relating to the use of the CMM, this study will examine the key practices of the CMM and evaluate how they apply to the broader processes and functions operating within a typical IS operation. To identify what the “typical” IS processes and activities might be we referred to an earlier, more comprehensive framework of IS processes: the Information Systems Management Architecture (ISMA). The ISMA was developed in the early 1980’s by researchers working for IBM to identify generic business processes of an IS organization and assess the effectiveness of those processes (Van Schaik, 1985).

In the following section, we will describe the CMM and ISMA models and their original intentions. Next, we map the CMM key process areas and practices against the process definitions and descriptions of the ISMA. The paper concludes with a discussion on the potential ramifications of using the CMM which is focused on systems development to improve quality within the IS organization.

**Brief Overview of the CMM**

The CMM “is a framework that describes the key elements of an effective software process (Paulk, 1995, pg. 4).” Its goal is to provide guidance identifying the few key issues most critical to improving the ability of organizations to meet goals for cost, schedule, functionality, and product quality. The CMM is organized into five “maturity levels” which define the evolutionary plateaus leading to the achievement of a mature software process (Paulk, 1995). These maturity levels roughly parallel those defined by Philip Crosby (1979) in his Quality Management Maturity Grid (QMMG). The QMMG and the CMM define a mature organization as one that follows a disciplined process consistently and where the necessary infrastructure exists to support the processes. This in turn leads to process capability, which is the inherent capability of a process to produce planned results. The underlying premise of the CMM is that the quality of a software product is largely determined by the quality of the process used to develop and maintain it (Paulk, 1995).

The five software process maturity levels identified by Humphrey (1989) and incorporated into the CMM consist of the Initial, Repeatable, Defined, Managed, and Optimizing levels. During the Initial level the software process is characterized as ad hoc and chaotic. Few processes are defined and success depends on individual efforts and heroics. At the Repeatable level, basic project management processes are established to track cost, schedule, and functionality, and the necessary discipline is in place to repeat earlier successes on projects with similar applications. This discipline is further strengthened at the Defined level, when software processes are documented, standardized, and integrated into a standard software process for the organization. During the Managed level the organization begins collecting detailed measures of software process and product quality, and both the software process and products are quantitatively understood and controlled. Finally, at the Optimizing level continuous process improvement is enabled by the quantitative feedback from the process and from piloting innovative ideas and technologies.

Each maturity level of the CMM indicates a level of process capability and is composed of several key process areas. These key process areas, in turn, identify a cluster of related activities, referred to as key practices, that when performed collectively achieve a set of goals considered important for enhancing process capability (Paulk, 1995). The CMM assigns key process areas to each “maturity level” except the Initial level (see Table 1). At the Initial level the organization, by definition, has few consistent processes, most are ad hoc and are not repeated consistently by everyone. At the Repeatable maturity level the CMM identifies the Requirements Management, Software Project Planning, Software Project Tracking and Oversight, Software Subcontract Management, Software Quality Assurance, and Software Configuration Management key process areas. At the Defined level, it describes the Organization Process Focus, Organization Process Definition, Training Program, Integrated Software Management, Software Product Engineering, Intergroup Coordination, and Peer Reviews key process areas. During the Managed maturity level the key process areas identified are Quantitative Process Management and Software Quality Management. Finally, at the Optimizing level the CMM describes the Defect Prevention, Technology Change Management, and Process Change Management key process areas.

**Brief Overview of the ISMA**

The ISMA was developed from research beginning in the early 1970’s on Business Systems Planning by IBM. The ISMA separates IS processes into three distinct missions: the Development Mission, the Service or Operations Mission, and the Consultation Mission (Van Schaik, 1985). Eleven IS management process groups are defined within these three missions which are further subdivided into 42 process areas. For each of these process areas the ISMA defines its function and purpose, followed by a description of the activities performed within it. In addition to identifying and describing the typical IS processes, the ISMA framework can be used to assess the effectiveness of those processes.

Depending on which IS processes have been effectively implemented, the ISMA designates the IS organization as being at one of five “Phases of Growth”. The definition of these phases is based on Nolan’s six “EDP Stages of Growth” model (Nolan, 1979). These ISMA phases consist of the
Related Content

An Artifact Switching Model for User Acceptance of eBooks
[www.igi-global.com/article/artifact-switching-model-user-acceptance/76875?camid=4v1a](www.igi-global.com/article/artifact-switching-model-user-acceptance/76875?camid=4v1a)

A Two-Tier Approach to Elicit Enterprise Portal User Requirements
[www.igi-global.com/chapter/two-tier-approach-elicit-enterprise/18288?camid=4v1a](www.igi-global.com/chapter/two-tier-approach-elicit-enterprise/18288?camid=4v1a)

Diffusion of Adoption of Facebook for Customer Relationship Management in Australia: An Exploratory Study

Problems and Pitfalls in the Evaluation of Adaptive Systems
[www.igi-global.com/chapter/problems-pitfalls-evaluation-adaptive-systems/18179?camid=4v1a](www.igi-global.com/chapter/problems-pitfalls-evaluation-adaptive-systems/18179?camid=4v1a)