Brady Field

The Capability Maturity Model for Software

**Important points**

“After decades of unfulfilled promises about productivity and quality gains from applying new software methodologies and technologies, organizations are realizing that their fundamental problem is the inability to manage the software process.” [345]

“The CCM provides software organizations with guidance on how to gain control of their processes for developing and maintaining software and how to evolve toward a culture of software engineering and management excellence.” [346]

“Maturity implies a potential for growth in capability and indicates both the richness of an organization’s software process and the consistency with which it is applied in projects throughout the organization.” [347]

“The priorities in the CCM, as expressed by these levels, are not directed at individual projects. A project that is in trouble might well prioritize its problems differently that the taxonomy given by the CCM.” [348]

“A defined software process contains a coherent, integrated set of well-defined software engineering and management processes. A well-defined process includes readiness criteria, inputs, standards and procedures for performing the work, verification mechanisms (such as peer reviews), outputs, and completion criteria. Because the software is well defined, management has good insight into technical progress on the project.” [349]

“There is chronic waste, in the form of rework, in any system simply due to random variation. Organized efforts to remove waste result in changing the system by addressing ‘common causes’ of inefficiency.” [350]

“Unprecedented systems complicate the picture since new technologies and applications lower the process capability by increasing variability. Even in the case of unprecedented systems, the management and engineering practices characteristic of more mature organizations help identify and address problems earlier than for less mature organizations. In some cases a mature process means that ‘failed’ projects are identified early in the software life cycle and investment in a lost cause is minimized.” [351]

“Because of the diverse uses of the CCM, it must be decomposed in sufficient detail that actual process recommendations can be derived from the structure of the maturity levels. This decomposition also indicates the key processes and their structure that characterize software process maturity and software process capability.” [352]

“The adjective ‘key’ implies that there are process areas (and processes) that are not key to achieving a maturity level. The CCM does not describe in detail all the process areas that are involved with developing and maintaining software. Certain process areas have been identified as key determiners of process capability, and these are the ones described in the CCM.” [353]

“The specific practices to be executed in each key process area will evolve as the organization achieves higher levels of process maturity.” [354]

“Goals summarize the key practices of a key process area and can be used to determine whether an organization or project has effectively implemented the key process area. The goals signify the scope, boundaries, and intent of each key process area. Satisfaction of a key process area is determined by achievement of the goals.” [355]

“The CCM identifies the characteristics of an effective software process, but the mature organization addresses all issues essential to a successful project, including people and technology, as well as process.” [356]

**Disagreements**

I could not find anything specific to disagree with in this article. It was well written and more informational than opinionated. I read it twice looking for something to disagree with. The only thing I feel could have been done better is to have included more references in the text.

**Questions**

I understood everything in the article.