

## Sorting Out Six Sigma and the CMM

David N. Card

Over the past few years, two approaches to process improvement have gained widespread acceptance. You'll find the Six Sigma approach broadly applied across major corporations such as Motorola and General Electric, while others such as Lockheed Martin and Telecordia have focused on the Capability Maturity Model. Some corporations have adopted both. The purpose of this column is to help sort out the relationship between these two approaches.



### What Is Six Sigma?

Six Sigma provides a generic quantitative approach to improvement that applies to any process. The specific measures and analyses employed must be tailored to the domain of processes under study. You'll find Six Sigma commonly described as a philosophy that implies the body of knowledge required to implement it. The curriculum of the Six Sigma Institute's training program provides the best definition of the relevant body of knowledge. Jerome Blakeslee Jr. provides a definition of Six Sigma in his July 1999 *Quality Progress* article, "Implementing the Six Sigma Solution":

*Basically, it is a high-performance data-driven approach to analyzing the root causes of business problems and solving them. It ties the output of a business directly to marketplace requirements.*

*At the strategic level, the goal of Six Sigma is to align an organization keenly*

*to its marketplace and deliver real improvements (and dollars) to the bottom line. At the operational level, Six Sigma's goal is to move business product or service attributes within the zone of customer specifications and to dramatically shrink process variation—the cause of defects that negatively affect customers.*

The name, Six Sigma, derives from a statistical measure of a process's capability relative to customer specifications. The Six Sigma notion of capability does not have the same meaning as the definition of capability implied in the title of the CMM. The latter deals with the maturity of practices, not results.

### What Is the CMM?

The CMM describes the principal disciplines, or key process areas, that an effective software engineering organization must master. These include basic software engineering and management as well as improvement practices. The implementation of key process areas is organized into five sequential steps or levels. Level 1 does not require mastery of any processes. Levels 2 and 3 require mastery of 13 key processes and focus on defining a software organization's basic technical and management processes. Levels 4 and 5 address controlling and improving those defined processes. Five key process areas make up Levels 4 and 5. In *The Capability Maturity Model: Guidelines for Improving the Software Process* (Addison Wesley, 1994), Mark C. Paulk and his colleagues showed how two of these key process areas largely parallel Six Sigma:

*Quantitative Process Management involves establishing goals for the performance of the project's defined software process, which is described in the Integrated Software Management key process area, taking measurements of the process performance, analyzing those measurements, and making adjustments to maintain process performance within acceptable limits.*

*Defect Prevention involves analyzing defects that were encountered in the past and taking specific actions to prevent the occurrence of those types of defects in the future. The defects may have been identified on other projects as well as in earlier stages or tasks of the current project. Defect prevention activities are also one mechanism for spreading lessons learned between projects.*

The other Level 4 and 5 key process areas go beyond a Six Sigma program's usual scope. *Software Quality Management* addresses estimating the output quality of a series of software processes, each of which might be subject to control and improvement. *Technology Management* provides a proactive focus on technology as a means of improvement. *Process Change Management* helps provide coordination for improvement activities across a software organization.

#### Six Sigma's Weaknesses

Six Sigma evolved from experience in manufacturing. A manufacturing process has inherent visibility. For example, you can observe the flow of materials, and opportunities for measurement are usually obvious. By contrast, software development is an intellectual process that must be made visible (that is, documented) before you can measure and manage it. Six Sigma doesn't specifically address this situation. The CMM focuses on defining processes through Levels 2 and 3.

Six Sigma relies on trained per-

sonnel to find processes that need improvement and act accordingly. Training alone often doesn't change behavior. Important process elements might or might not get attention. The CMM requires a systematic search for and prioritization of improvement opportunities.

Six Sigma doesn't have a formal connection to ISO 9000 or other certification programs for organizations, although proposed updates to ISO 9000 incorporate Six Sigma techniques. In the long term, business benefits are what count and Six Sigma stresses them. However, the CMM levels and certification can provide near-term incentives and convenient milestones for checking an implementation's progress.

Six Sigma focuses internally on learning from current process experience. It does not consider, systematically, how externally developed technology might revolutionize or even eliminate a process. The CMM requires a specific technology focus, which can be essential to survival in a dynamic business such as software.

#### CMM's Weaknesses

The CMM document describes process control in terms of means, standard deviations, special causes, common causes, and so on. However, significant elements of the CMM user community resist the obvious need for statistical methods to achieve Levels 4 and 5. No such ambiguity exists in the Six Sigma approach.

The CMM states that measures and processes should be selected based on their relationship to strategic business plans and goals, but this requirement often gets overlooked. Software engineering process teams often set software goals without considering business goals. Six Sigma

stresses the need to link process improvement to business performance.

The CMM does not describe the body of knowledge required to implement it. Many implementations merely restate CMM requirements. Thus, the improvement effort might not incorporate the full range of measurement and analysis techniques that Six Sigma promotes.

Although the Software Engineering Institute certifies CMM assessors, no comparable program exists for CMM implementers. Assessors must be familiar with the CMM, not the underlying body of knowledge—for example, statistical process control. The Six Sigma Institute provides competence certification for relevant subject matter.

#### Common Elements of Six Sigma and CMM

Both approaches focus on reducing defects as the primary method of improvement. However, Six Sigma provides a relatively broad definition of a defect, and the CMM acknowledges the possibility of focusing on other dimensions for improvement.

Both approaches stress a quantitative, fact-based approach to decision-making that relies heavily on measurement. Successful introduction of measurement often requires cultural changes that neither approach specifically addresses. To some degree, the Six Sigma approach minimizes this by introducing these techniques opportunistically, addressing each process element (using CMM terms) individually.

Both approaches draw from the same toolkit of statistical process control techniques (control charts, pareto charts, cause-effect diagrams, and so on), although Six Sigma might introduce more advanced methods such as design of experiments, when appropriate. Unfortunately, the CMM's intent to encourage statistical methods sometimes gets lost.

Both Six Sigma and the CMM have drawn broad criticism—but such complaints are largely based on how these concepts have been implemented in specific organizations or promoted within the community,

**Six Sigma focuses internally on learning from current process experience.**

rather than fundamental problems in the approaches. The effective application of statistical techniques to software engineering has been amply demonstrated (for example, see the *Proceedings of the Software Engineering Process Group Conference*, Software Engineering Institute, 1999). Nevertheless, no single technique applies in every situation—the analyst needs a toolkit.

The concepts of CMM Levels 4 and 5 and Six Sigma are synergistic. For example, Six Sigma training helps build the skills necessary to address the CMM's Quantitative Process Management and Defect Prevention requirements. The CMM provides a structure that helps ensure the systematic application of Six Sigma techniques. The CMM helps explain how to adapt Six Sigma techniques to software processes.

The situation, however, merits two notes of caution. First, some organizations rated at CMM Level 5 might not have implemented Six Sigma techniques because they opted to instead satisfy CMM requirements with ad hoc analysis methods. You can't assume that a Level 5 rating implies Six Sigma competence. Second, a software engineering organization that has not made significant progress toward defining its key process elements (for example, CMM Level 3) should only introduce Six Sigma techniques on a small-scale pilot basis. There are too many examples of large investments in extensive Six Sigma training for personnel working in unstructured environments—investments that led to little return.

Six Sigma offers another path toward measurable improvement for CMM Level 3 organizations. Instead of pursuing full CMM Level 4 and 5 maturity, an organization can just address the subset that constitutes Six Sigma—depending on its business needs for CMM certification. Moreover, the Six Sigma toolkit provides the basic technology necessary for continuous improvement beyond the CMM Level 5 certificate. 

David N. Card is the chief scientist for quantitative methods at the Software Productivity Consortium; card@software.org.

## CORNING

Corning Science and Technology, the company's research and development organization, uses an entrepreneurial approach that has been helped by substantial investments in recent years. Research, development and engineering facilities have doubled in size since 1997. Today, more than 1400 scientists and engineers work around the globe, linking technology to business strategy in telecommunications, information display, semiconductor manufacturing, environmental quality and advanced life sciences.

### Sr. Software Engineers/Project Leaders

(Ad Code # AD/IEEESoftware/281)

We are seeking talented software engineers, controls engineers and project leaders with a BS, MS or Ph.D. degree and a minimum of 1 year of appropriate experience. Project Leaders will have C++ and related RTOS, C & Assembly using DSPs, Embedded controllers, control theory and algorithms.

### Embedded Software & Systems Design Engineers

(Ad code #: AD/IEEESoftware/993406)

- \* System architecture definition and distributed real-time embedded system design
- \* Object-oriented design and analysis
- \* Systems level integration and testing
- \* Design using Rational Rose and ObjecTime
- \* Code generation using C++, VB, and C for deployment on Windows NT, VxWorks, or other RTOS
- \* Code generation using C and Assembly using digital signal processors.
- \* Development within the software engineering process as outlined in the SEI CMM.

### Software Quality Assurance & Test Engineers

(Ad code #: AD/IEEESoftware/993407)

- \* QA engineers work with various software development organizations within Research & Development to implement standard software development process, monitor software development projects and suggest improvement in the software development processes and software product quality
- \* You'll identify SQA and testing processes, tools, and best known methods to help resolve issues and assist software teams in implementing solutions.
- \* Test Engineers will be required to provide in-depth alpha testing of applications and software
- \* You must have experience in generating test cases/plans, Windows 95/98/NT/2000, Graphical tools, the ability to troubleshoot, test hardware/software & identify problems.
- \* May have designing and coding automated tests at the component and functional level using C++ with 3 years of coding experience in C/C++, WIN32api, and Multithreading.

### Senior Process Control Engineers

(Ad code #: AD/IEEESoftware/574)

The Process Control Systems Group is seeking candidates with a BS or MS in EE, CE, ME, ChemE or related with 5+ years of related experience in any of the following technical areas:

- \* Fuzzy logic and neural network based controllers
- \* Adaptive control (i.e. using a reference model function)
- \* Predictive controllers
- \* Nonlinear control
- \* Control systems with data and sensor validation, pattern recognition, fault classification, and multivariable control applications
- \* Database design and Data mining / analysis tools
- \* Active X control, COM, and DCOM
- \* Standard COTS software
- \* Control instrumentation hardware and software
- \* Machine and motion controls
- \* High-level computer languages such as C, C++, Java, and VB
- \* Real Time operating systems and real time control
- \* Intranet control and WEB based GUI design

Want to change the world? Start by re-inventing yourself at Corning Inc. Visit us at [www.careers.corning.com](http://www.careers.corning.com). All positions are located at our Corning, NY facility.

All resumes must reference the associated Position Title and #, and should be submitted in MS WORD format via e-mail to: [Mandicottf@corning.com](mailto:Mandicottf@corning.com). Or mail to: Frank Mandicott, Corning Recruiting, Attn: Ad Code # \_\_\_\_\_, SP-FR-02 MOD1, Corning, NY 14831. If you have any questions, please contact Frank Mandicott directly at (607) 974-6737. Principals only please. Corning is an equal opportunity employer with a strong commitment to workforce diversity.

INNOVATING AT THE SPEED OF LIGHT