# Blending CMM and Six Sigma to Meet Business Goals

**Mala Murugappan and Gargi Keeni,** *Tata Consultancy Services*

Capability maturity models such as SW-CMM provide organizations a framework for process improvement, helping them identify which process areas need attention to reach a certain maturity level. However, many organizations find it difficult to match the process improvement goals with customer expectations and to predict and measure the capability in schedule, effort, and quality. Six Sigma provides the means to explicitly address these issues, thereby giving organizations a firm basis on which to sustain the process improvement efforts.

At Tata Consultancy Services (TCS), we blended Six Sigma concepts and various Software Capability Maturity Model key process areas[1] into our own quality management system. We structured our QMS around SW-CMM Level 2 and Level 3 requirements—including quantitative process and product measurements and analysis, process improvements for defect prevention, and process optimization—and reinforced these with Six Sigma concepts, such as initiating Six Sigma projects for continuous improvements. In this way, TCS-QMS provides an organized structure for quality improvement and customer focus, paving the path to CMM Level 5.

Here we describe our approach and its benefits, explaining how we leverage Six Sigma to gain statistical insight into the alignment of software goals and customer needs. We also discuss how our method helps initiate the cultural change required to ensure continuous improvement and effective implementation of SW-CMM's Level 4 and 5 KPAs.

**Integrating Six Sigma with SW-CMM provides the framework for process improvement and customer focus by aligning process improvements goals with customer expectations.**

## TCS's CMM-based process framework

TCS is India's largest IT enterprise, providing IT and business consulting services to organizations in government, business, and industry in India and abroad. Our services are varied and straddle many different industries, including finance and banking, insurance, telecommunications, transportation, retail, manufacturing, pharmaceuticals, and utilities. Over time, we've integrated various quality concepts into the TCS-QMS to meet changing business goals.[2]

- 1993: Documented software development practices, maintenance, and conversion to conform to ISO 9000

 

- 1995: Reinforced all improvement initiatives under the Tata Business Excellence Model umbrella (along the lines of Malcolm Balridge National Quality Award
- 1996: Adopted SW-CMM to establish defined and consistently implemented mature software processes throughout the organization
- 1998: Applied Six Sigma methods to quantitatively analyze problems with products and processes to ensure customer satisfaction
- 2000: Adopted the People-CMM to integrate people practices to sustain SW-CMM practices and ensure employee satisfaction[3]

Currently, 15 of TCS's 17 development centers in India are operating at SW-CMM Level 5. Using TCS-QMS, we've established an infrastructure in these development centers that institutionalizes effective software engineering and management processes across projects.

The TCS-QMS ensures customer satisfaction by

- Designing processes that ensure product and service quality
- Integrating quality control activities into software development
- Stressing quality assurance through defect prevention techniques
- Using metrics to manage processes and quality
- Introducing new technologies to ensure continuous process improvements

The TCS-QMS process framework has several key features. First, it provides a foundation for effective project management by tracking cost, schedule, quality, and functionality. We've established process discipline to repeat earlier successes on projects with similar applications. We also use estimation guidelines, checklists, procedures, project management review meetings, and automated project-tracking tools[4] to satisfy the goals of Level 2 KPAs.

Second, TCS-QMS provides an architecture for organization-wide knowledge sharing. This architecture includes a dedicated software process engineering group and process owners to institutionalize process improvements,[5] along with well-defined software development life cycles, training programs, tailoring guidelines, service-level agreements within groups, and product peer review.[6,7] A process assets library fosters institutionalization and sharing of best practices.

Third, TCS-QMS emphasizes data-driven management of software product and process quality. It does this using different statistical techniques and defines targets and tolerances for various metrics. It also emphasizes the analysis of process performance against capability baselines and encourages preventive actions to eliminate process instability.[8]

Finally, TCS-QMS facilitates continuous process improvement with quantitative feedback from the process itself. It empowers team members to pilot innovative ideas and technologies aimed at process and quality improvements. We also use statistical tools like Pareto analysis and cause-and-effect analysis to identify the source of process problems and to identify and prioritize improvements.

## Blending CMM and Six Sigma

The concepts of SW-CMM Level 4 and Level 5 and Six Sigma are synergistic. Six Sigma is a customer-centric, data-driven approach that focuses on reducing process variation, centering—making the process mean coincide with the process target—and optimizing the development process. While SW-CMM provides the basic process infrastructure to systematically apply Six Sigma techniques,[9] Six Sigma helps to build the knowledge and skills necessary to address SW-CMM's key requirements, including quantitative process management, software quality management, defect prevention, technology change management, and process change management. Together, Six Sigma and CMM help organizations improve marketplace competitiveness and achieve their business goals.

**Together, Six Sigma and CMM help organizations improve marketplace competitiveness and achieve their business goals.**

### Process management: The Six Sigma way

Using the Six Sigma approach has added many new facets to TCS's process management. Based on the Six Sigma principle, we designed our metrics program to translate customer needs into operational measurements. We also carry out detailed analyses of process trends by clearly distinguishing common cause variations from special cause variations. Finally, we eliminate the ambiguities in

## Table 1

### Quality function deployment

| First House of Quality | Product requirements | | | | |
|---|---|---|---|---|---|
| Customer requirements | Rating | Object-oriented design | Use cases | Traceability | Coding standards |
| Reliability | 10 | 9 | 9 | 9 | 3 |
| Maintainability | 10 | 9 | 3 | 9 | 9 |
| Portability | 7 | 9 | 3 | 3 | 3 |
| Testability | 5 | 3 | 9 | 9 | 1 |
| Total of rating × requirement | | 258 | 186 | 246 | 146 |

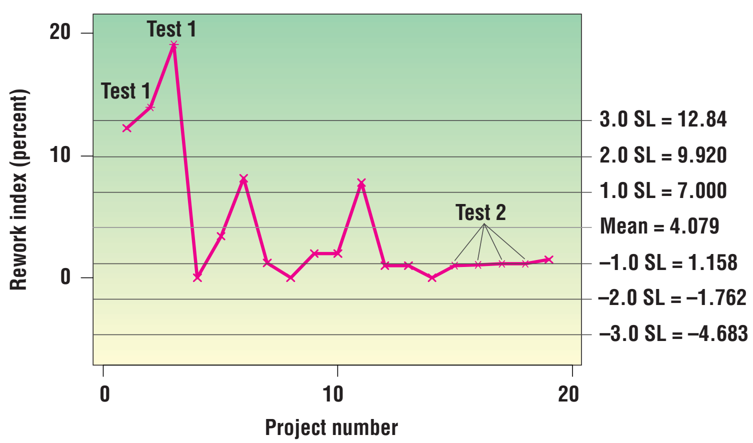


**Figure 1. Control chart patterns for a sample process. The process failed two tests: Test 1 identified points two and three as outside the 3.00 sigma limit (SL); Test 2 identified points 15–18 as 1.00 sigma from the center line.**

calculating process capability using the sigma capability, which considers both process variability and process centering objectives.

***Customer-centric metrics program.*** Our project teams identify the software process and product metrics to track based on Quality Functional Deployment, a quality tool that offers a systematic way to break down customer needs into manageable actions. Teams typically carry out QFD with customers, translating customer requirements into project requirements. They then rank process characteristics in order of importance and identify those that are critical to quality. Table 1 shows a QFD house of quality matrix, which identifies a project life cycle's critical quality characteristics;[10] the column totals represent the sum of the products of each customer requirement rating and product requirement.

The team first translates the first column's customer requirements into the product requirements (object-oriented design, use cases, traceability, and coding standards). The team then rates the relationship between product characteristics and customer requirements based on the relationship's strength (9 is a strong relationship, 3 a medium relationship, and 1 a weak relationship). In the second house of quality, the team further analyzes product requirements with high technical importance to determine the design features needed to meet these requirements.[11]

Based on the QFD results, the team selects in-process metrics that indicate progress toward the customer-specified improvement objectives.

The object here is to

- Align measurements with business priorities
- Benchmark process and product performance against client-established targets, operational requirements, and quality objectives
- Improve the processes to attain these defined performance limits
- Facilitate the goal of continuous improvement

***Metrics analysis and management.*** The concept of process variability forms the heart of process metrics analysis.[12] Process variation can be partitioned into two components:

- *Natural process variation,* frequently called common-cause or system variation, is the naturally occurring fluctuation or variation inherent in all processes.
- *Special cause variation* is typically caused by some problem or extraordinary occurrence in the system.

We must reduce both the common and special cause variation to make the process more predictable and produce the best output. Software organizations typically identify special causes through the presence of points outside the control limits of $\pm 3\sigma$. TCS piloted the Six Sigma approach of analyzing control chart patterns to identify potential process instability indicators. Once we identify an unstable pattern, we find the assignable cause and correct the process. Figure 1 shows a process tested for the points outside the 3$\sigma$ limit (Test 1) and the points that are 1$\sigma$ from the center line (Test 2).

The team analyzed the special causes of these patterns and found that the Test 1 failures resulted from tight schedules, which created more rework after internal reviews. They corrected this by loading project teams more uniformly. The second set of test results stemmed from team members whose prior project experience created less rework; the test results are a good indication that they shared knowledge across the team.

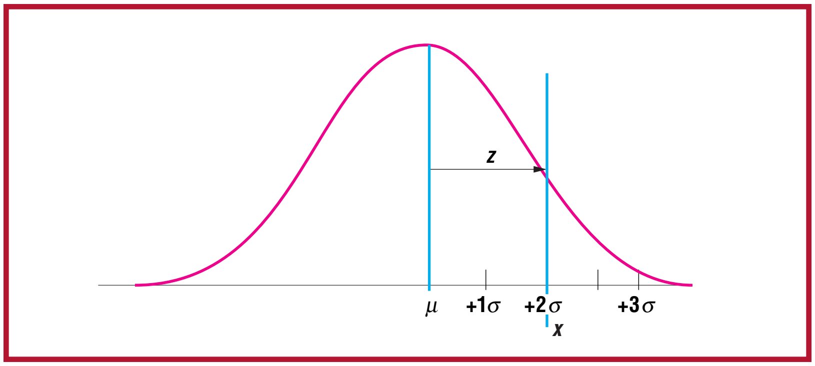


**Figure 2. Process capability in terms of *Z* capability: *x* is the specification limit, $\mu$ is the mean, and $\sigma$ is the standard deviation.**

***Process capability calculations.*** Six Sigma typically monitors the process using control charts, which compare control limits with specification limits to determine process capability. We measure process capability in terms of capability indices $C_p$, $C_{pk}$, and $C_{pm}$,[12] which are simplified measures that quickly describe the relationship between process variability and the specification limits' spread. Like many simplified measures, capability indices do not completely describe the process; they are best used to compare process capability.

In terms of $Z$ capability (sigma capability), process capability is the number of standard deviations that fit between the mean and the specification limit; this unit corresponds to defect probability. We compute it as

$$Z = (x - \mu) / \sigma$$

where $x$ is the specification limit, $\mu$ is the mean, and $\sigma$ is the standard deviation (see Figure 2).

The essential value of process capability analysis is its ability to predict the percentage of process characteristics or products that the process will produce within specification limits. Graphically, $Z$ capability is the distance, measured in standard deviations, from the process mean to the specification limit. In other words, the $Z$ capability is the area under the normal curve up to the specification limit.

Tracking process capability in terms of $Z$ ensures both our process variation reduction and process centering objectives. It also helps us baseline the process capability and clearly compare process performance with specified targets. $Z$ is a simple metric for measuring a process's capability to produce defect-free products.

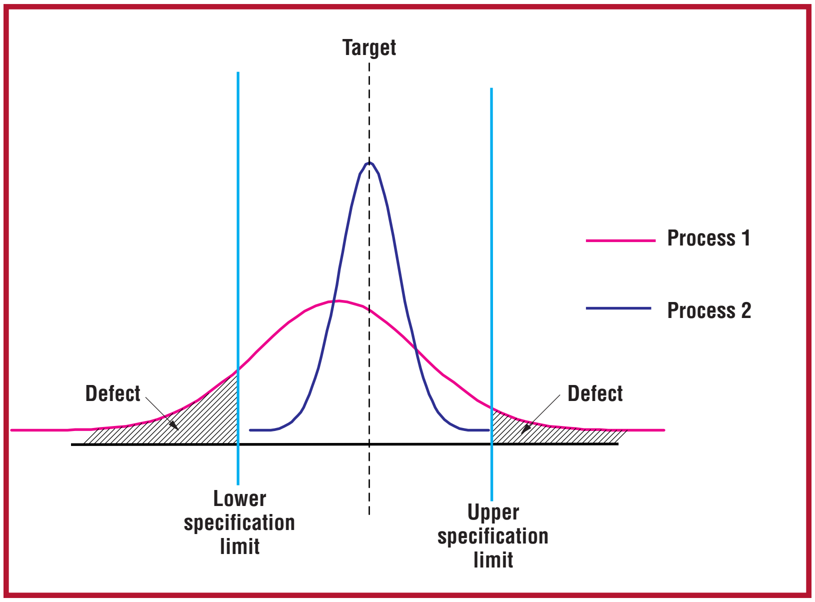


**Figure 3. Use of *Z* capability for predicting process capability. The shaded area denotes defect probability. For a Six Sigma process, the error probability is 0.0000034 (that is, there are only 3.4 defects per million delivered units). For a 3$\sigma$ process, the probability of error is 0.0668 (66,800 defects per million units).**

Figure 3, for example, shows two processes. Process 1 has a mean below the target, has large variation, and produces many defects outside the specification limits. In contrast, process 2 is centered on the target, has less variation, and has no defects outside the specification limits. Compared with process 2, process 1 has less $Z$ capability (as measured by the percentage of normal curve within specification limits). Thus, calculating process capability in terms of $Z$ not only helps project managers understand process dynamics but also helps them estimate the relative percentage of acceptable and unacceptable characteristics that the process is producing on an ongoing basis.

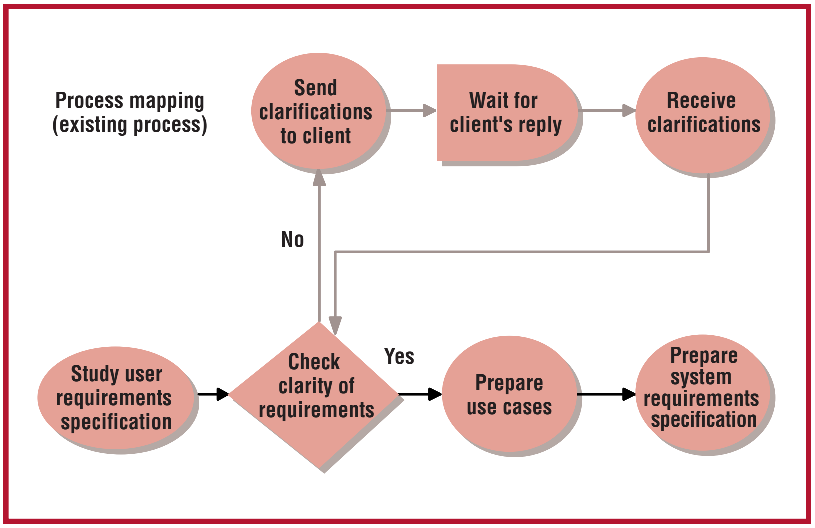


**Figure 4. Process mapping. Implemented as a CMM Level 3 process, process mapping breaks down a process so that teams can identify gaps in standard software practice.**

### Continuous improvement

Blending Six Sigma and SW-CMM helps us ensure our goal of continuous improvements. Six Sigma provides the foundation to define, measure, analyze, improve, and control the processes,[13] whereas methods such as process mapping and Failure Modes and Effects Analysis (FMEA) help us understand process defects and prioritize improvement actions.

As Figure 4 shows, we've implemented process mapping as a CMM Level 3 practice to help us understand gaps in an organization's standard software process by breaking the process down into its constituent elements. Process mapping shows the ordering of process elements and interfaces, and interdependencies among elements. We also use process mapping as a CMM Level 5 process improvement practice, breaking down complex processes so that teams can see them in their entirety. This helps them identify process delays and loops, identify problem spots and improvement targets, and determine the data to collect.

As Table 2 shows, we also use FMEA to identify improvement areas. Here, it helps teams

- Identify how a process can fail
- Determine failure mode severity
- Predict the probability of failure causes occurring
- Find deficiencies in the process control plan
- Prioritize improvement actions

A team assigns potential failure modes a severity rating from 1 to 10, with 10 having the highest impact on customers. It assigns failure causes a 1 to 10 occurrence rating, with 10 representing the highest likelihood of occurrence. Finally, the team assigns a 1 to 10 detection rating, with 10 indicating the lowest likelihood of failure mode or cause detection.

The team then combines these ratings to produce a risk priority number. It prioritizes these numbers for the top few problems, identifies improvement actions, assigns responsibilities, and determines time frames for completion. Once it has implemented the actions, the team carries out FMEA again to ensure that it's reduced the failure mode's risk.

A more common way to evaluate risk is to assess a risk's impact and frequency of occurrence, then calculate the risk exposure accordingly. The FMEA approach differs in that it considers the process controls' effectiveness and helps a team prioritize the risks. Teams can use FMEA effectively for risk management for SW-CMM Level 3 and for defect prevention at SW-CMM Level 5.

## Benefits to the organization

Six Sigma and SW-CMM complement each other and together can help an organization meet its process improvement goals.

**Table 2**

**Failure mode and effects analysis**

| Potential failure modes | Potential causes | Current controls | Risk priority number | Recommended action |
|---|---|---|---|---|
| Improper requirements analysis | Lack of understanding of requirements | Review of software requirements specification | 200 | Training on software requirements management tools |
| Bugs in code | Lack of programming skills | Code review | 180 | Coding standards |
| Poor code review efficiency | Untrained reviewers | Testing | 160 | Code review checklist |
| Improper project estimation | Lack of knowledge | Project management reviews | 140 | Estimation guidelines |

The benefits of blending the methods are illustrated by results from a TCS development center. As Figure 5 shows, the center used SW-CMM and Six Sigma concepts to reduce its in-process failure cost from 5 to 1 percent, thus reducing the cost of quality.

The center also had several process and technology improvements for cycle-time reduction and productivity improvement. For example, it implemented

- A statement of work to capture all project requirements
- The tollgate approach for project execution
- Defect prevention checklists
- Estimation guidelines
- Standard libraries
- Guidelines and methodologies
- Error-proofing tools

The center's process capability for product quality improved from 96 percent in October 1999 to 100 percent in September 2000. In terms of sigma capability, the center's product quality is currently 5.85 (that is, clients reported no defects in software products). Process capability for on-time delivery improved from 2.85 in October 1999 to 4.50 in October 2001. Finally, as Figure 6 shows, the Six Sigma techniques have reduced the center's schedule slippage variation from the 20 to –20 percent range to the 2 to –2 percent range. Applying these methods saved the center US$700,000 over a three-year period. We subsequently integrated these practices in the TCS QMS and deployed it across the organization.

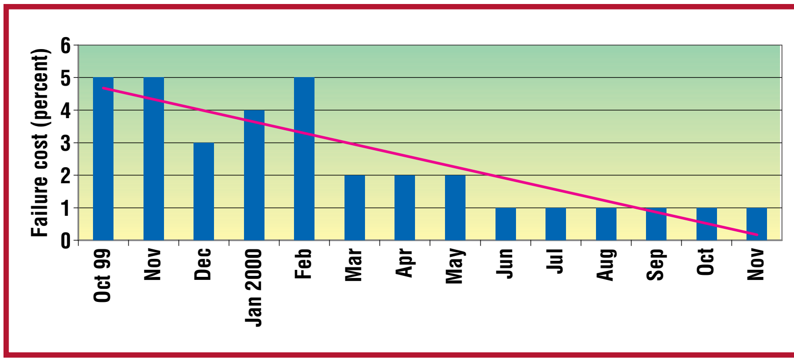


**Figure 5. Trend of failure costs. Using SW-CMM and Six Sigma concepts, the organization reduced its failure costs to 1 percent.**

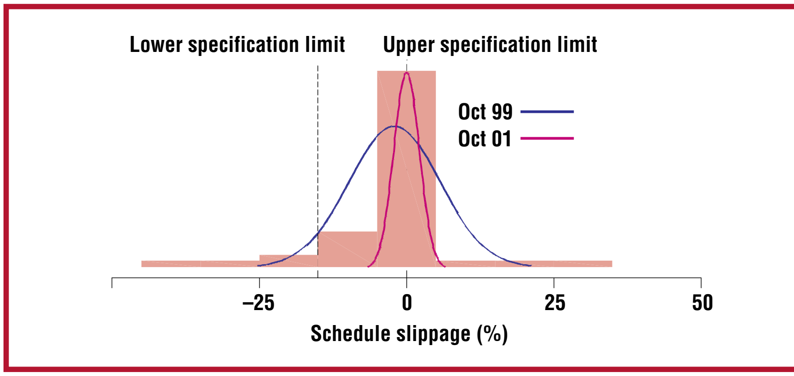


**Figure 6. Process capability analysis for schedule slippage.**

We designed the TCS QMS to ensure the quality of products and services and to emphasize continuous improvement. By breaking down the customer expectations into process- and product-level attributes, the Six Sigma program offers precise operational definitions for improvements. This in turn helps teams translate the organization's strategic goals into tactical objectives, which become the process improvement initiatives in Six Sigma projects. Teams can then feed the derived improvements into the organization's CMM-based process framework, which helps define and institutionalize the improvement actions into disciplined and mature processes.

We've deployed these blended CMM and Six Sigma concepts in development centers in India through the TCS QMS. The next step is to institutionalize the Six Sigma methodology of Define, Measure, Analyze, Improve, and Control (DMAIC) to all process improvement initiatives and to transfer these best practices in TCS's Global Development Centres throughout the world.

## References


1. M.C. Paulk et al., "Capability Maturity Model Version 1.1," Software Engineering Institute, Carnegie Mellon Univ., CMU/SEI-93-TR-24, DTIC No. ADA263403, Pittsburgh, Pa., 1993.
2. G. Keeni, "The Evolution of Quality Processes at Tata Consultancy Services," *IEEE Software*, vol. 17, no. 4, July/Aug. 2000, pp. 79–88.
3. G. Keeni, B. Curtis, and C. Kubicki, "Lessons Learned in Achieving People CMM Level 4 at Tata Consultancy Services," *Proc. US Software Eng. Process Group Conf.*, 2002, Software Engineering Institute, Carnegie Mellon Univ., 2002; http://seir.sei.cmu.edu/seir/frames/body2.map.P-CMM.html.
4. G. Keeni and D. Biswas, "Evolution of an Automated Quality Management System," *Proc. Int'l Symp. Future Software Technology* (ISFST 2001), Software Eng. Assoc., Tokyo, 2001, pp. 349–358.
5. G. Keeni, A. Chandra, and S. Dutta, "Structure of the Quality Group for Effective Software Process Improvement," *Proc. Software Process Improvement Conf.*, 2000; www.uni-paderborn.de/cs/ag-schaefer/Tagungen/2000/spi00/spi00.pdf.
6. R. Radice et al., "One on One Inspection," *Proc. Software Technology Conf.*, STC Online, Logan, Utah, 2001; www.stc-online.org/stcdocs/stc2001/178/index.htm.

## About the Authors

**Mala Murugappan** is the Master Black Belt at Global Engineering Development Centre in TCS, Chennai, India. A certified quality analyst, her interests include process and quality improvement, and Six Sigma quality. She has undergone Six Sigma training for Black Belt and carried out several Six Sigma projects for quality and productivity Improvement. She has supervised more than 50 Six Sigma projects including Green Belt and Black Belt projects, and has been trained by SEI on Software CMM. She received her BEng in mechanical engineering from the College of Engineering, Guindy, Chennai, and her MS in machine design from the Indian Institute of Technology, Chennai. Contact her at Global Engineering Development Centre, 185-188, Lloyds Road, Chennai 600 086 India; mmala@chennai.tcs.co.in.

**Gargi Keeni** is vice president at Tata Consultancy Services, where she heads the quality consulting practice and assists organizations in achieving their software process improvement goals. Her research interests include software process improvement and quality management systems. She is principal architect of the Integrated Project Management System, which provides all the necessary operational metrics from the project level to the organizational level. She has more than 20 years' experience in software project management, software tools development, and software process improvement. Under her leadership as Corporate Quality Head, TCS's 15 development centers were assessed at Software CMM Level 5, and TCS was the first organization to be assessed at Level 4 of the People CMM v2.0. She is an authorized Software CMM Lead Assessor, People CMM Lead Assessor, Lead Appraiser for SCAMPI, an examiner for the Tata Business Excellence Model, and a Certified Quality Analyst. She has a PhD in nuclear physics from Tohoku University, Sendai, Japan. She is a member of the Computer Society of India and IEEE. Contact her at Tata Consultancy Services, SDF building, Salt Lake Electronics Complex, Sec-V, Block EP and GP, Kolkata 700 091, India; gargi@ieee.org.


7. R.P. Nandivada et al., "The 9 Quadrant Model for Code Reviews," *Proc. Asia Pacific Conference on Quality Software*, IEEE CS Press, Los Alamitos, Calif., 2000, pp. 188–193.
8. R. Radice, R. Sokhi, and P. Suresh, "The Journey to Level 4," *Proc. European Software Eng. Process Group* (ESEPG 99), Software Technology Transition, 1999; www.stt.com/TCSjourney.PDF.
9. D.N. Card, "Sorting out Six Sigma and CMM," *IEEE Software*, vol. 17, no. 3, May/June 2000, pp. 11–13.
10. R. Prasad and G. Keeni, "Integrated Demanded Quality Deployment and QFD," *Proc. 13th Quality Function Deployment Symp*., QFD Inst., Ann Arbor, Mich., 2001.
11. M. Murugappan and G. Keeni, "Incorporating Voice of Customer in the Six Sigma Way," *SEPG Conf. Tour in Asia-Pacific*, 2002; www.softwaredioxide.com/channels/conview.asp?id=6693.
12. S. Sytsma and K. Manley, "Quality Tools Cookbook," June 1999; www.sytsma.com/tqmtools/tqmtoolmenu.html.
13. M. Murugappan and G. Keeni, "Quality Improvement: The Six Sigma Way," *Proc. Asia Pacific Conf. Quality Software*, IEEE CS Press, Los Alamitos, Calif., 2000, pp. 248–257.


For more information on this or any other computing topic, please visit our Digital Library at http://computer.org/publications/dlib.