

Measure

Advice, Trends, Opinions and Analysis from Gartner Measurement

Special Edition 2001
Capability Maturity Model

In This Issue

Describing the Capability Maturity Model	1
The Relevancy of the Capability Maturity Model to Commercial IS Organizations	3
Maximizing Improvement Based on the Capability Maturity Model	6
The Benefits of the Capability Maturity Model for Application Development	9

Describing the Capability Maturity Model

The Capability Maturity Model® (CMM®) for software is a widely accepted set of guidelines for developing high-performance software organizations. Watts Humphrey and his colleagues at IBM developed the original concept for the CMM in the early 1980s. Humphrey determined that the quality of an application was related directly to the quality of the process used to develop it. To improve application development processes, Humphrey wanted to implement the W. Edwards Deming continuous-improvement cycle (i.e., plan — do — check — act). However, application developers had been installing improved development methods and technologies for a decade without gaining much improvement.

Humphrey observed that improved software development practices did not survive unless an application development (AD) organization's behavior changed to support them. His unique insight was that AD organizations had to remove impediments to continuous improvement in a specific order if they were to succeed. Consequently, Humphrey designed the process maturity framework, an evolutionary path to help AD organizations increase the capability of their AD processes in five stages (see Figure 1).

During the evolution through the five maturity levels, development practices are transformed from an ad hoc, undisciplined state into disciplined processes capable of predictable results. At its core, the CMM is a unique model of organizational development and change. As an AD organization progresses from one level to the next, its culture is transformed through the evolutionary improvement of its development processes. Each maturity level is characterized by the implementation and institutionalization of several clusters of practices (i.e., process areas) that contribute to the development capability achieved at that level (see Figure 2).

Figure 1
The Five Maturity Levels of the Capability Maturity Model

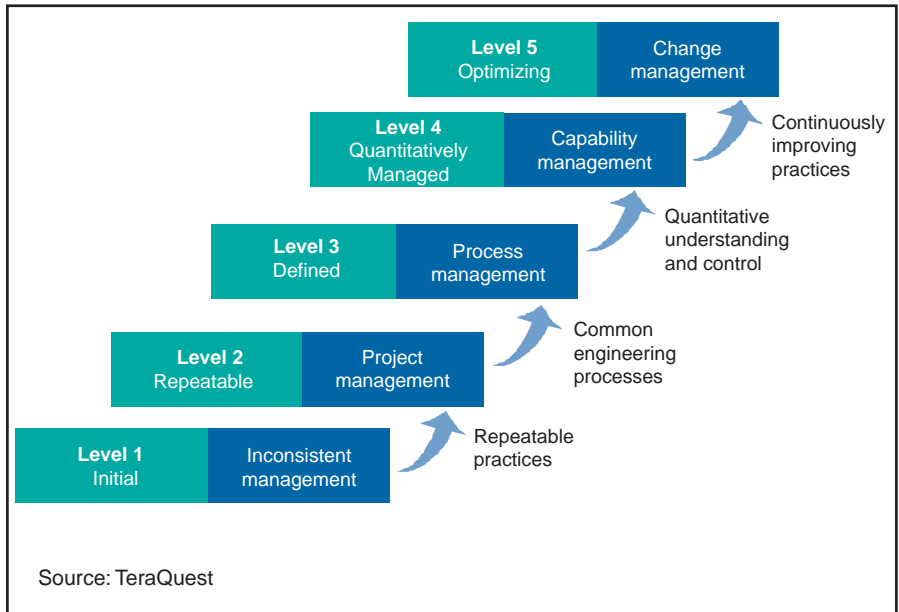


Figure 2
Capability Maturity Model Process Areas

Maturity Level	Improvements Implemented
5. Optimizing	<ul style="list-style-type: none"> • Develop change infrastructure • Evaluate and deploy improvements • Eliminate causes of defects
4. Quantitatively Managed	<ul style="list-style-type: none"> • Manage processes quantitatively • Establish capability baselines
3. Defined	<ul style="list-style-type: none"> • Establish improvement infrastructure • Identify required software processes • Define common software processes • Deploy and manage processes • Collect process-level data • Provide organization-wide training • Coordinate with non-software groups
2. Repeatable	<ul style="list-style-type: none"> • Manage requirements • Plan and track projects • Manage suppliers • Manage product configurations • Measure projects • Assist and assure policy compliance
1. Initial	No required processes

Source: TeraQuest

The Maturity Levels

Level 1: Initial — At this base level, application development practices and results are inconsistent. Development processes rarely are defined, and sound practices often are sacrificed to meet unreasonable schedules. Although developers are capable of performing their assignments, they do so through individualized methods that show little consistency across the AD organization. Frequently, project management is weak and does not protect developers from the disruption created by unreasonable commitments or excessive requirements changes. Essentially, the Level 1 AD organization lacks the capability to meet commitments consistently.

Level 2: Managed — It is critical first to establish a stable environment that facilitates the repetition of successful practices. Thus, Level 2 focuses on developing the capabilities of project managers to plan achievable commitments and establish control of requirement baselines and product configurations. Although projects may use different methods or practices, the environment must be stabilized to support their performance. AD organizations with Level 2 capabilities deliver their applications on schedule without having to survive on heroes and constant overtime.

Level 3: Defined — After projects can repeat successful practices, AD organizations identify best practices from different projects. Subsequently, these procedures are integrated into a common AD process and deployed across the AD organization. Hence, a strong organizational culture emerges at Level 3 based on a common process that covers all the important elements of AD. Once all projects use tailored versions of a common AD process, an AD organization can begin comparing results, sharing lessons learned and transferring people more easily among projects. When an AD organization can begin estimating from historical data resulting from common processes, it is much easier to achieve targets for cost, functionality and scheduling.

Level 4: Quantitatively Managed — Having established common AD processes, an AD organization then can develop statistical capability baselines that characterize the expected results from performing these procedures. These baselines

provide a profound, quantitative understanding of the capability of development processes and the causes of variation in their performance. By managing the performance of its development processes statistically, an AD organization can predict and control project outcomes much earlier in the course of a project. Quantitative management allows greater empowerment of project teams and increased predictability of results for project management.

Level 5: Optimizing — Despite the achievement of predictable results, targeted business objectives may not be achieved. At the highest level of maturity, an AD organization continuously evaluates the capability of its processes to pinpoint areas requiring the greatest improvement. Continuous improvements can be developed opportunistically by deploying the results of lessons learned, or they can be produced proactively by evaluating new development methods, processes or technologies for potential adoption. Ultimately, a Level 5 AD organization estab-

lishes an infrastructure for supporting continuous change management as a fundamental, integral component of its overall development process.

The CMM achieved widespread adoption because it broke the cycle of “silver bullets” and “big bangs” that characterized earlier attempts to improve AD. At each stage of its evolutionary improvement path, the CMM implemented an integrated collection of management and development practices that built on the infrastructure the AD organization had established at earlier maturity levels. These processes then became the new foundation for more sophisticated processes at the next level. Consequently, as an AD organization achieved the next level of maturity, the culture moved one step further away from its initial frenzied state toward an environment of professionalism and continuous improvement.

Source: Dr. Bill Curtis, TeraQuest — a Gartner alliance partner
Writer: Carolyn LeVasseur, Gartner

The Relevancy of the Capability Maturity Model to Commercial IS Organizations

Since its introduction in the early 1990s, the Capability Maturity Model® (CMM®) has steadily gained acceptance by software organizations. Starting with its initial implementation in the defense industry, the frame-

work has spread across sectors and is poised for significant penetration into commercial IT (see Figure 1). Given its roots, it is important to ask whether the CMM is relevant to the problems facing IS organizations today.

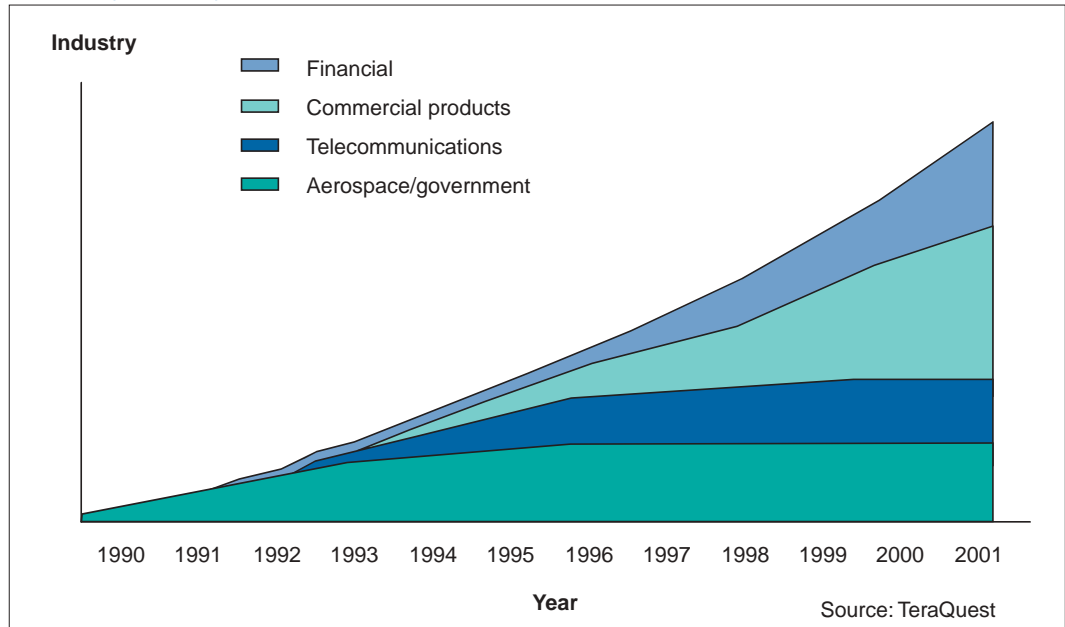
Clearly, the answer is “yes.” For example, the CMM can be used as a means to drive customer alignment, manage external service providers or increase productive output.

Recent Gartner studies demonstrate that the average IS organization devotes approximately one half of its IT budget to developing new applications and supporting/maintaining its existing applications development (AD) portfolio (see *2001 IT Spending and Staff Survey Results*, Gomolski, et al., R-14-4158, 19 Sept. 01). The sheer

magnitude of this investment necessitates processes that will ensure a return on this investment. In addition, it is critical that these procedures be part of an agile culture — one that can react to changes in an increasingly competitive global marketplace.

Gartner's AD benchmark data reflects the volatility caused by rapidly changing businesses; the average project undergoes significant last-minute project changes driven by end users and IT groups (see Figure 2). Gartner's AD TOP (i.e., Technology, Organization and Process) model measures average project stability at 2.8 out of 5.0, which indicates significant uncertainty of requirements throughout the life of the average project. Maintaining alignment with end users (i.e., in IT and non-IT) is

Figure 1
Capability Maturity Model Adoption



nearly impossible without some level of structure in development processes. A lack of structure results in changes that spiral out of control, releases that must be rescheduled and mounting end-user frustration. Through the management foundation provided at Level 2 and the common, leveraged processes at Level 3, the CMM provides structure and an evolutionary approach for incremental improvements.

Probably the most visible end users of the CMM in IT have been the external service providers. By 2003, Gartner analysts project that the proportion of AD outsourcers and service providers at CMM Level 2 will triple, and those at Level 3 will double (0.8 probability) (AD Scenario, Vecchio, 14H,

Sym11, 10/01). Such growth in CMM adoption and resultant organizational maturity have been driven to some degree by outsourcers using the CMM as a way to discriminate among, and qualify, providers. For others, especially offshore service providers, the CMM has enabled the establishment of credibility by offering a maturity-level rating as an indicator of AD capability.

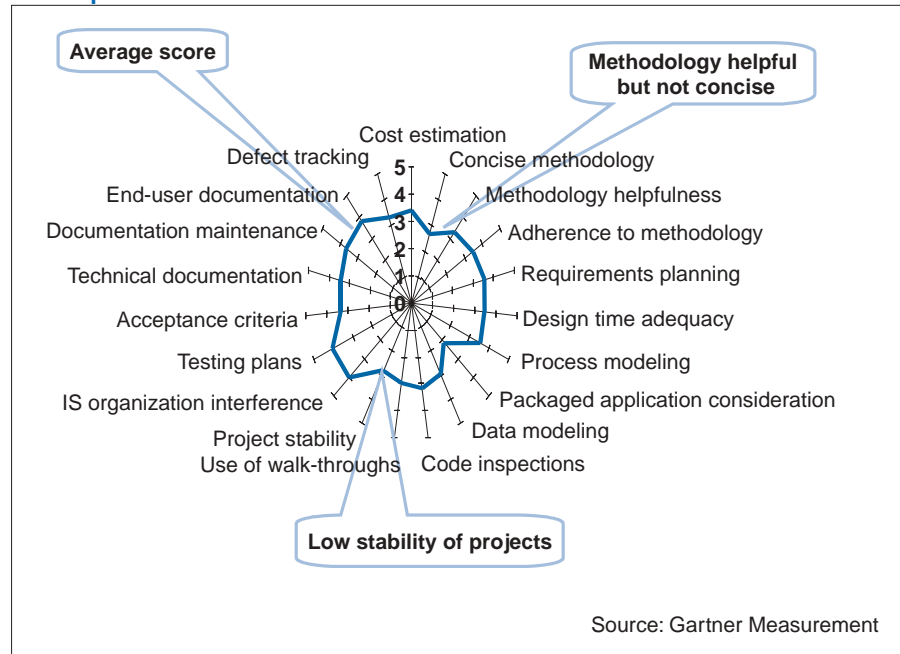
Despite the frequent focus on service provider maturity levels, outsourcers should not lose sight of their own internal capability levels. To achieve competencies equal to service providers, IT/AD management should implement their own process improvement programs aimed at CMM Level 2. These capabilities will enable them to manage projects, control requirements

and configurations, assure quality and track project progress. Moreover, industry evidence continues to mount that immature outsourcing organizations can render more mature vendors less effective in meeting project requirements.

Increasingly, as the economic downturn drives a focus on internal efficiencies vs. external “revolutionary” changes, IS organizations are concerned about improving productivity. Furthermore, with stagnated or reduced staffing levels and the increased business demands on IT, CIOs must improve the maturity of their processes to increase the efficiency and effectiveness of their existing personnel. IS organizations attaining increased maturity levels manifest substantial reductions (e.g., by two-to-three times) in rework, thus increasing their capacity to tackle additional projects. To reduce the time spent handling defects, which can consume a substantial portion of IT effort, it is important to establish processes that enable earlier detection. According to numerous industry studies, IS organizations achieving at least a Level 2 stage of maturity demonstrate a significant decrease in effort and numbers of problems reaching customers.

Although IS organizations strive for greater maturity and common processes, it is vital to ensure

Figure 2
Development Process



that processes remain agile and facilitate a quick reaction to changing business needs. Existing methodologies are considered reasonably helpful but not very concise (see Figure 2). Thus, competitive advantages accrue from processes that foster creative approaches while maintaining management control. A frequent misconception of the CMM is that it specifies a methodology for execution that is cumbersome (i.e., does not permit innovation). In fact, the CMM does not specify a methodology at all. Instead, it provides goals that establish a process foundation for progressive levels of maturity. Proper execution of development life cycles, such as rapid application development and extreme programming, are consistent with the fundamentals at the heart of the CMM.

The CMM's growing acceptance stems from its applicability to real problems in AD organizations and the published results of achievements by those using it to guide process improvements. Although results for commercial IS organizations only now are starting to be published, current IT business drivers and the CMM's goals will generate gains similar to those experienced in other software-intensive organizations. These potential gains alone make the CMM highly relevant to the future of commercial IS organizations.

Source: Mike Epner, TeraQuest — a Gartner alliance partner
Bob Solon, Gartner
Writer: Carolyn LeVasseur, Gartner

Maximizing Improvement Based on the Capability Maturity Model

Although every sound enterprise pursues continuous improvement, those that are responsible and best in class require that improvements demonstrate measurable benefits. At times, those benefits are long term and take several years of organizational change. To justify investing current resources in an improvement program, management usually must cite near-term worth. The Capability Maturity Model® (CMM®) for software provides a framework for software process improvement that concurrently supports long-term and short-term value.

Long-term benefits from using the CMM for improvement often are associated with major changes (e.g., transitioning an IS organization from one level of maturity to the next). To achieve success, goals can be targeted according to the CMM focus at each level. For example, CMM Level 2 reflects a level of maturity in which an organization has established effective project management within projects so that project commitments are met. This enables the organization to attain goals such as delivering to promised schedules within some small margin of error, managing the impact of requirements changes and maximizing the match of functionality to expectations.

With Level 3 performance, an IS organization employs a common process framework for running its projects. Such a framework enables it to maintain useful project history databases, leverage lessons learned across projects, share staff among projects and employ best practices. Higher levels of maturity enable predictable business results and the use of process performance measures to understand overall process capability.

Depending on its current maturity level, an IS organization can

set appropriate goals for its improvement program in several ways. For example:

- Strategy-focused balanced scorecards
- The techniques of Practical Software and Systems Measurement (see www.psmc.com)

Figure 1 shows a sample strategy map for an IS organization using process improvement to support shorter project cycles and smaller features as well as enable more-frequent product updates and a higher revenue stream.

Figure 1
Sample Strategy With Software Process Improvement Objectives

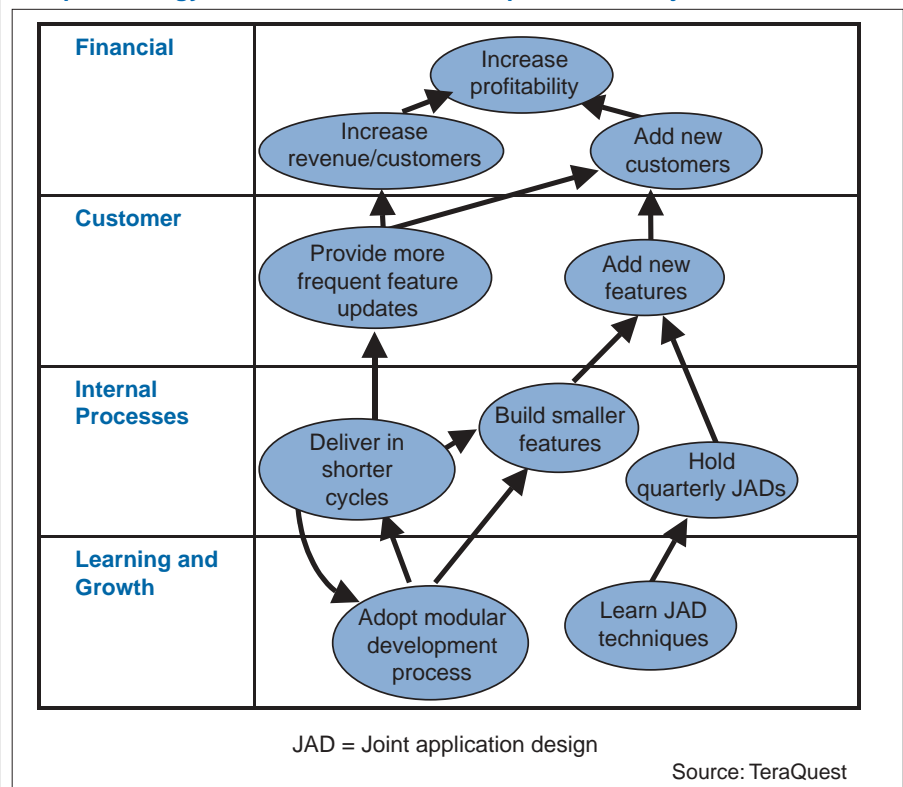


Figure 2
Sample Software Process Improvements in Balanced Scorecard

Area	Measure	Target
Financial	<ul style="list-style-type: none"> Profit Mean revenue/customers Number of new customers 	<ul style="list-style-type: none"> 25% before tax +30% more than last year +5% more than last year
Customer	<ul style="list-style-type: none"> Number of updates/year Number of features added per year 	<ul style="list-style-type: none"> +25% more than last year +25% more than last year
Internal Processes	<ul style="list-style-type: none"> Average cycle time (i.e., months) Average feature size (i.e., FPs) Product delivery rate (FPs per calendar month) 	<ul style="list-style-type: none"> -50% of last year -50% of last year +25% more than last year
Learning and Growth	<ul style="list-style-type: none"> Percentage of projects using new modular process Percentage trained in JAD 	<ul style="list-style-type: none"> 100% by end of 2Q 100% of target group by March 1

FP = Function point
JAD = Joint application design

Source: TeraQuest

Associated example measures are outlined in Figure 2.

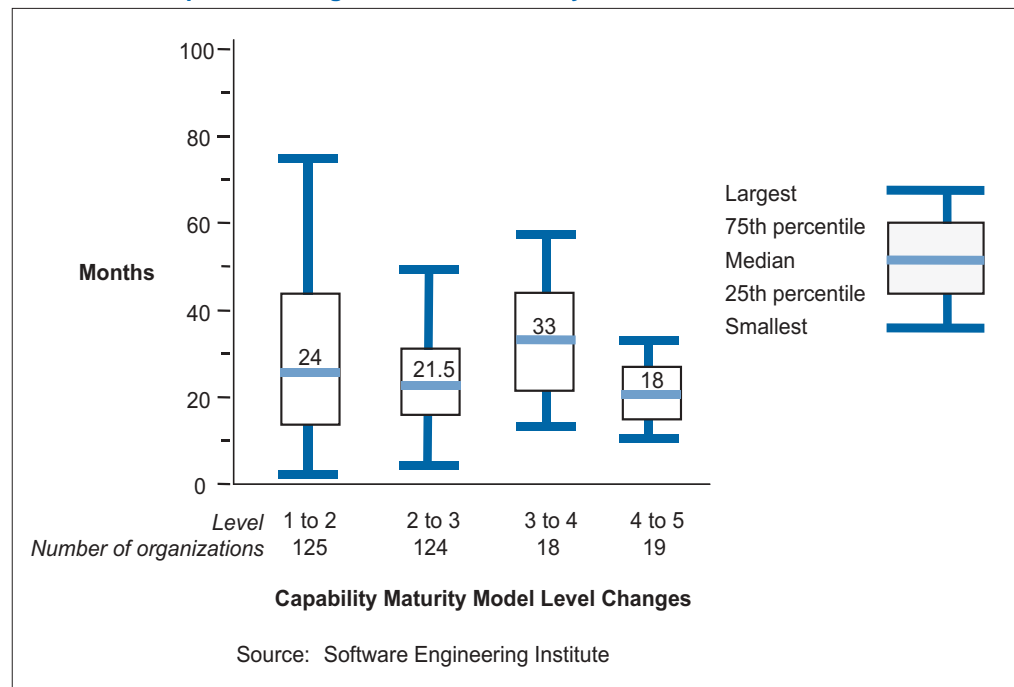
Attaining such long-term benefits takes time and organizational patience. The transition time to move from one CMM maturity level to the next often spans about two years (see Figure 3). In today's fast-paced business world, IT managers require quick results and visible value. To hasten the payback from improvement, it is critical to implement small, incremental changes or ones that are localized to projects. For IS organizations starting a CMM-based improvement journey, a worthwhile approach focuses on projects within the overall process im-

provement plan and introduces small, but useful, changes in each project.

When undertaking an improvement initiative, deciding on appropriate changes to a project

can be approached in several ways. Certainly, it is important to ascertain the degree of impact of a given project on the overall organizational strategy. If, for example, a project's contribution required it to provide its

Figure 3
Usual Time Required to Progress to Next Maturity Level



deliverables in a shorter-than-normal cycle time, it might need a new work breakdown structure (WBS). The project team could be trained in that WBS as part of its project launch workshop.

Alternatively, project teams may take the lead and target a specific aspect of their work requiring substantial improvement. If, for example, no team estimation process existed, a project launch workshop could teach a Delphi estimating process that then could be applied to their project.

In each of the above examples, the project team implements a change in the work that impacts positively the whole IS organization's capability to manage projects. Two other elements are essential for effective project improvement work. As a project proceeds, it is crucial to provide a mentor who can answer questions about the new processes. Furthermore, the greatest value is realized when a mentor works with the team to consolidate the lessons learned for incorporation into future projects.

The source of a mentor may vary (e.g., a senior project manager, a quality assurance

analyst or someone from a process group). Whatever their role, it is vital that the mentor be part of a team that coordinates the overall process improvement effort. Such coordination provides lessons learned across projects, identifies useful examples and best practices for teams to implement and encourages the IS organization to gather resulting process assets into a common location. For areas in which an IS organization does not yet have any process assets, the coordinating team can develop those for use within the projects, thus encouraging the evolution of shared practices.

As new processes are implemented in current projects, the short-term benefits quickly become apparent. Accordingly, management views the effort as justified, and the short-term value is established. In addition, an economic benefit accrues because the improvement work is performed as part of the development project work. Thus, the bulk of the funding for an IS organization's improvement activity is incorporated into the project work and is not funded separately as an extra expense. However, some investment is required external to any project.

Funding must be available for the mentors and the overall coordination effort needed to plan the organizational improvement activities. For most IS organizations focused on continuous improvement and the long-term benefits realized from the accumulation of project-focused improvements, this usually amounts to a minor 1 percent to 3 percent.

The CMM provides guidance for long-term and short-term improvement efforts. For project teams focused on a particular area, the CMM identifies best practices and offers helpful guidance on training, measures and verification activities. For the coordinating improvement team, the CMM provides guidance on which areas should be pursued first to establish a workable overall plan that builds organizational capability.

Source: Joyce Statz, TeraQuest — a Gartner alliance partner
Writer: Carolyn LeVasseur, Gartner

The Benefits of the Capability Maturity Model for Application Development

During the past decade, software organizations conducting improvement programs guided by the Capability Maturity Model® (CMM®) have reported gains in productivity, quality, time to delivery, accuracy of cost and schedule estimates as well as product quality. Because aerospace enterprises were the first to adopt the CMM, they were the source of much of the early data on improvement benefits. Fortunately, improvement results are finally emerging from other application developers, primarily in the United States and India.

As an AD organization proceeds from one maturity level to the next, the range of benefits from its improvement activities and processes increases substantially. Since improvements at each maturity level solve different sets of problems, different benefits emerge at each level. These benefits are summarized in Figure 1.

Many of the initial benefits from CMM-based improvement programs result from eliminating rework. Low-maturity AD organizations usually estimate shorter development schedules, which pleases executives. However, the evidence indicates that they actually take longer than projects with lengthier estimated schedules. This happens because they are forced into extensive rework on top of the actual effort required to produce the application on a disciplined schedule.

Enterprises such as Hewlett Packard, Raytheon, TRW, and NASA have reported that when their processes were immature, between 30 percent and 50 percent of their development effort was spent on fixing mistakes (i.e., on rework). In low-maturity AD organizations, developers or their managers too often commit to schedules that cannot be achieved without heroic effort. When rushing out code to meet unrealistic dead-

lines, developers cut corners and sacrifice sound development practices. Consequently, they generate scores of undetected mistakes. As developers integrate and test their applications, a “tsunami” of defects emerges and overwhelms the project with rework. With mounting defect levels, the project loses control of its schedule and budget.

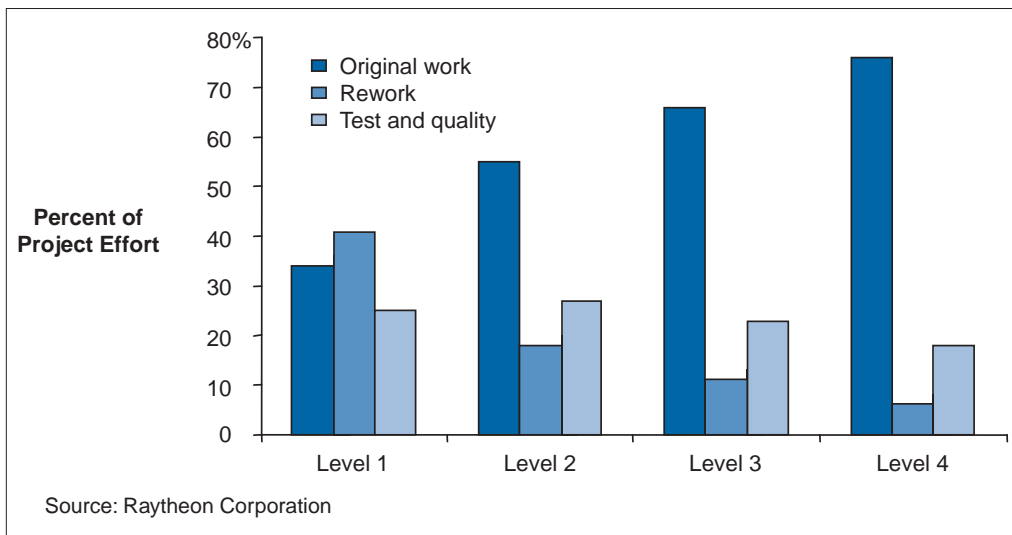
By achieving CMM Level 2, projects can set realistic expectations, commit to attainable deadlines and avoid the Level 1 “death marches” on nights and weekends that produce excessive defects. In addition, since project managers track progress against planned schedules, they can detect schedule slippages much earlier when it is still possible to take corrective action. Corrective actions often involve negotiating additional effort or reduced functionality. Consequently, Level 2 organizations meet release dates by trading off cost or functionality

Figure 1
Benefits of Capability Maturity Model Improvements

Level 5: Optimizing	Continuously targeting improvements required to meet business objectives
Level 4: Quantitatively Managed	Predictable results, knowledge of factors causing variance and reuse
Level 3: Defined	Meeting cost and functionality targets as well as improved quality
Level 2: Repeatable	Meeting schedule and reduced turnover resulting from less overtime
Level 1: Initial	No benefits: Inconsistency, schedule and budget overruns, and defective applications

Source: TeraQuest

Figure 2
Profile of Project Effort by Capability Maturity Model Level



targets. As they attain Level 2, many AD organizations report greater retention of developers by avoiding the overtime-laden death marches.

At Level 3, AD organizations use historical measures describing the performance of a common application development process as the basis for their estimations. As they gain experience with a common development process, they gain greater understanding of the interactions among schedule, effort and functionality. This learning process allows them to meet commitments in all three

dimensions simultaneously. In addition, because they have installed peer reviews and other defect-detection methods earlier in their development process, they continue to reduce defect rates and the resulting rework.

Figure 2 displays effort distributions collected by a Raytheon organization during the six years that it moved from Level 1 to Level 4. At each level, they continued to reduce rework and increase the percentage of effort involved in the original development activity. Even though the amount of effort involved in

testing, training, process improvement and other quality activities increased slightly as they achieved Level 2, they were able to discover ways to reduce the relative percentage of these quality costs over time. After reaching Level 3, their productivity increased by 230 percent from their starting baseline. By the time they moved to Level 4, they

were producing software for 40 percent of their original costs.

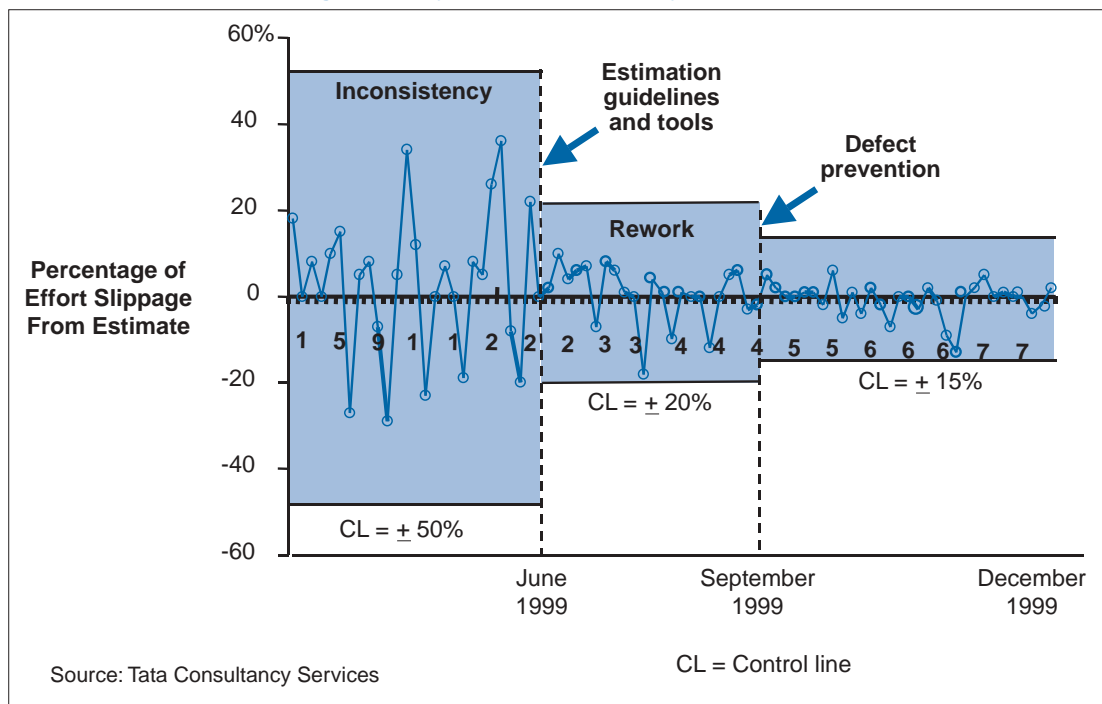
Similar results were achieved by an AD organization at Boeing when improvements were compared to baselines established at earlier levels (see Figure 3). The most noteworthy reduction in schedule variability occurred as they achieved Level 2, but reductions in defects, cycle time and costs were greatest as they moved to levels 3 and 4. Analogous results have been confirmed in case studies at L. M. Ericsson, Omron, Motorola, Telcordia Technologies and the United States Air Force. Furthermore, studies conducted at Carnegie Mellon University and the University of Southern California indicated that increases in organizational maturity produced significant gains in productivity and quality while reducing effort and time to delivery.

Figure 3
Percent Improvements Compared With Results at Previous Levels

Criteria	Level 1 → 2	Level 2 → 3	Level 3 → 4
Reduce defects	12%	40%	85%
Reduce cycle time	10%	38%	63%
Reduce cost	8%	35%	75%
Schedule variance	145%	24%	15%

Source: Boeing Corporation

Figure 4
Improvements in Estimating Accuracy at Tata Consultancy Services



As is evident in Figure 4, they identified two rounds of improvements: the first in which common estimating guidelines and tools were installed and the second in which the causes for specific types of product defects were eliminated. Consequently, these efforts were rewarded by succes-

Boeing, Omron and other enterprises have confirmed that IS organizations passing beyond Level 3 significantly increase their reuse of previously developed components. Software reuse has been the Holy Grail of application development productivity since the 1970s. However, reuse proved elusive in low-maturity environments in which developers knew they did not have time to properly design, test or document their applications. When AD organizations attain Level 3, the foundation required to support reuse (i.e., proper design, thorough testing and appropriate documentation) has been institutionalized. Having exhausted the source of productivity increases resulting from eliminating rework, AD

organizations passing beyond Level 3 find a new source of productivity gains from reusing trusted components. At Boeing, the number of projects with at least 30 percent reused code increased from less than 10 percent to more than 40 percent as they proceeded beyond Level 3.

High-maturity IS organizations (i.e., levels 4 and 5) have the ability to target specific problems, identify root causes and make improvements with predictable effects. For instance, Tata Consultancy Services' Level 5 offshore development center in Kolkata, India analyzed the root causes for variation in the accuracy of their effort predictions for work packages.

sively improved estimating accuracy. This pattern is typical of AD organizations engaged in continuous improvement.

Although the types of improvements experienced differ by the maturity level achieved, AD organizations at the lowest maturity levels begin experiencing returns on their investments (ROI) very early in their improvement programs. Despite reports of annualized ROI that have varied between \$4 and \$9 saved for every \$1 spent, executives still want to see early returns on their investments in process improvement. Fortunately, even as the earliest practices leading to Level 2 are being implemented, executives can recognize the growing

impact of improvements as projects become able to provide accurate accounting of their current state and anticipated risks. The increased capability of

project managers leads to improved results from the development of applications they manage. Ultimately, the journey through the succession of CMM levels is

underscored by short- and long-term benefits that constitute continuous improvement.

Source: Dr. Bill Curtis, TeraQuest – a Gartner alliance partner
Writer: Carolyn LeVasseur, Gartner

Dr. Bill Curtis Co-founder and Chief Scientist

Dr. Bill Curtis is Co-founder and Chief Scientist at TeraQuest, where he helps set business direction. Using his expertise in software process improvement, organization change, people management and technology evolution, Bill coaches management teams in how to orchestrate change, leads assessments of organization capability, and works with TeraQuest teams to craft appropriate programs for improving clients' organizational maturity.

Bill is a former Director of the Software Process Program at the Software Engineering Institute (SEI), where he led the project to produce the Capability Maturity Model® (CMM®) from the process maturity framework developed by Watts Humphrey. He is also the architect and primary author of the People CMM, an organization maturity model for attracting, developing, motivating, organizing and retaining an outstanding workforce. Bill heads a program to train Lead Assessors for the People CMM under an agreement with the SEI.

Prior to his work at the SEI, Bill directed research on advanced computing technologies at MCC, developed software productivity and quality measurement systems at ITT, and evaluated software engineering methods at GE. He has published several books and over 100 technical articles in the areas of software engineering and management. He is on the editorial boards of five technical journals and is a frequent and popular keynote speaker at software engineering conferences.

®Capability Maturity Model and CMM are registered in the U.S. Patent and Trademark Office.

Entire contents©2001 by Gartner, Inc. All rights reserved. Reproduction of this publication in any form without prior written permission is forbidden. The information contained herein has been obtained from sources believed to be reliable. Gartner disclaims all warranties as to the accuracy, completeness or adequacy of such information. Gartner shall have no liability for errors, omissions or inadequacies in the information contained herein or for interpretations thereof. The reader assumes sole responsibility for the selection of these materials to achieve its intended results. The opinions expressed herein are subject to change without notice.

Gartner