How CMM Impacts Quality, Productivity, Rework, and the Bottom Line

Michael Diaz and Jeff King
General Dynamics Decision Systems

The Software Engineering Institute's Capability Maturity Model® (CMM®) plays a major role in defining software process improvement (SPI) in many companies. The question of cost/benefit has come up frequently in organizations contemplating SPI activities. This article will explore various cost/benefit issues and examine performance results of various General Dynamics Decision Systems' projects with relation to its software process maturity. The quantitative data presented indicates CMM-based improvement yields dividends in terms of higher productivity and software quality. Each level of improvement significantly cuts defect density, improves productivity, and reduces rework.

General Dynamics Decision Systems supplies communications and information technology for military and government customers and employs approximately 1,500 engineers that design and build a wide variety of government electronic systems. Approximately 360 engineers are directly involved in software development.

The question of cost/benefit has come up frequently in organizations contemplating software process improvement (SPI) activities. This article will explore various cost/benefit issues and examine performance results of various General Dynamics Decision Systems' projects with relation to its software process maturity. It also discusses the implementation strategies put in place to achieve process improvement and other organizational goals; some "lessons learned" about process improvement are also presented.

CMM Overview

The Software Engineering Institute's (SEI) Capability Maturity Model® (CMM®) [1] plays a major role in defining SPI in many companies. The CMM consists of five levels of process maturity where each level has an associated set of key process areas (KPAs). At the initial Level 1 maturity, software projects rely on the skills and heroic efforts of individual engineers. There are no KPAs associated with Level 1. Firefighting is prevalent and projects tend to leap from one emergency to the next.

CMM Level 2, the Repeatable maturity level, has six KPAs associated with it. These KPAs relate to requirements management, project planning, project tracking and oversight, subcontract management, quality assurance, and configuration management. Projects under a Level 2 organization are repeatable and under basic management control.

At the Defined Level 3 maturity level, the software development organization now defines common processes, develops training programs, focuses on intergroup coordination, and performs peer reviews. The result is the development of tailorable software processes and other organizational assets so that there is a certain level of consistency across projects.

> *"Peer reviews have been widely recognized in the industry for being the single most important factor in detecting and preventing defects in software products."*

At the Managed Level 4 maturity level, the software development organization implements a quality and metric management program and monitors both project and organizational performance (i.e., establishes software process capability and variance measures using statistical process control charts).

At the Optimizing Level 5 maturity level, quantitative data are used for process improvement and defect prevention. In addition, technology changes are introduced and evaluated in an organized, systematic process.

Summary Results

General Dynamics Decision Systems has three software-engineering organizations: Integrated Systems, Information Security Systems, and Communication Systems. As of Nov. 16, 2001, all three had been externally assessed at CMM Level 5 using the CMM-Based Appraisal for Internal Process Improvement (CBA IPI). Our metrics and historical repository contain data from past Level 2, 3 and 4 programs within the Information Security Systems engineering organization, as well as our current Level 4 and 5 programs.

Rework, phase containment, quality, and productivity metrics are based upon history as well as the current measures of approximately 20 programs, each at various stages of the software life cycle.

At General Dynamics Decision Systems, every project performs a quarterly SEI self-assessment. The project evaluates each KPA activity with a score between one and 10, which is rolled up into an average score for each KPA. Any KPA average score falling below seven is determined to be a weakness. The SEI level for the project is defined as the level in which all associated KPAs are considered strengths, i.e., all KPA average scores are seven or above.

Table 1 summarizes the General Dynamics Decision Systems' improvement trends for rework, phase containment, predicted quality, and productivity by CMM level. Percent rework is a measure of the percentage of the project development time that was expended due to rework. Phase containment

Table 1: *General Dynamics Decision Systems Project Performance Versus CMM Level*

CMM Level	Percent Rework	Phase Containment Effectiveness	CRUD Density per KSLOC	Productivity (X Factor Relative)
2	23.2%	25.5%	3.20	1 x
3	14.3%	41.5%	0.90	2 x
4	9.5%	62.3%	0.22	1.9 x
5	6.8%	87.3%	0.19	2.9 x

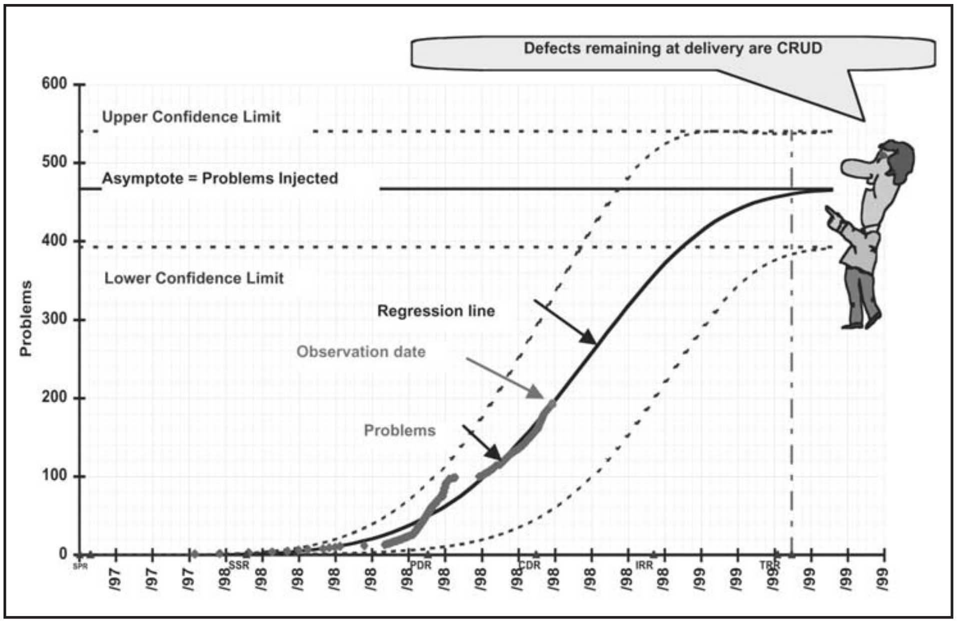


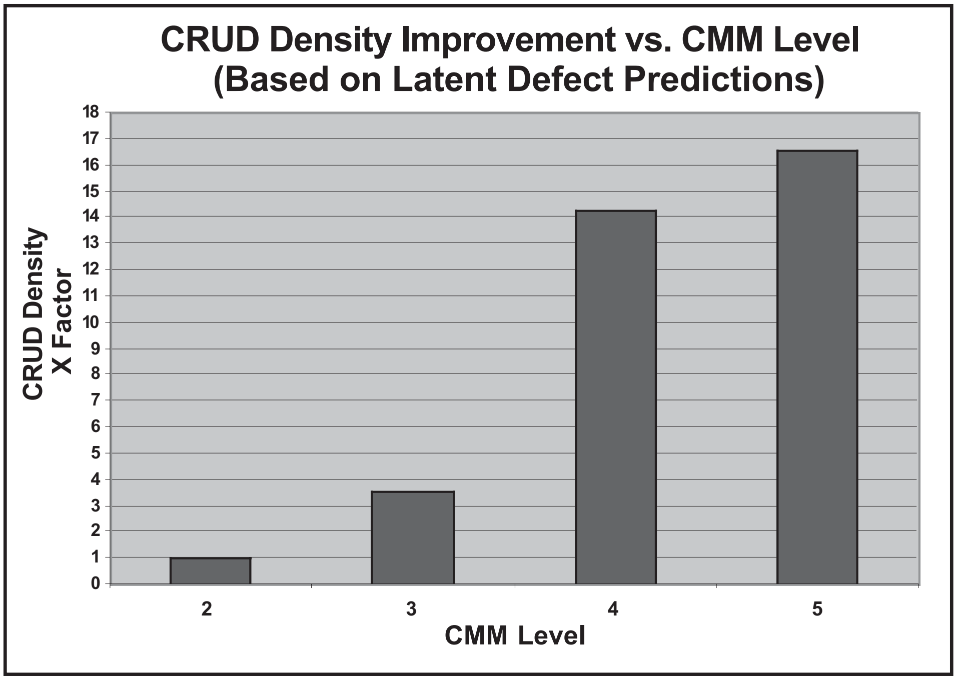
Figure 1: *Customer Reported Unique Defects (CRUD) Prediction Chart*

effectiveness is a measure of defect containment within the phase in which it was created. Higher phase containment is equivalent to early detection of defects within the same phase in which it was created. Predicted quality is defined as the number of latent defects or Customer Reported Unique Defects (CRUD) per thousand source lines of code (KSLOC). Productivity is displayed in X factor terms that are defined as the productivity average of all programs within a certain CMM level divided by the productivity average of all Level 2 programs. The quality, rework, and productivity performance for each program is obtained from General Dynamics Decision Systems' internal metrics and categorized by CMM level as determined by each project's internal self-assessment.

Detailed Metric/Results Analysis

This section will discuss each General Dynamics Decision Systems' metric collected (i.e., percent rework, phase containment, quality, and productivity) and discuss the improvement results with relation to CMM maturity level. As will be discussed later, specific improvement results are not entirely attributable to increasing CMM maturity levels since the organization has put into place initiatives in cycle time and quality improvement above and beyond the SEI CMM.

Figure 2: *Quality Versus CMM Level*



1. Quality Metric

At General Dynamics Decision Systems, post release software quality is defined as the number of predicted latent defects per thousands of delivered source instructions. Latent defects are predicted based upon the rate of new problems discovered during development. The total number of problems introduced in a software product is the sum of problems detected during development and the latent defects remaining at product release.

The total number of software problems introduced in a software product can be estimated by using historical defect density data from similar projects and tracking problems found early in the development cycle. A method to predict problems throughout the development cycle is given by Arkell and Sligo in "Software Problem Prediction and Quality Management [2]." Latent defects or CRUD are defects in the delivered product. CRUD is not experienced during software product development but it can be estimated.

Future problems can be predicted from the pattern of problems already detected by examination (peer review or inspection), by testing, and by using work products already examined or tested. The cumulative number of problems detected over time tends to follow an S-shaped curve.

The number of problems remaining to be discovered (CRUD) is depicted in Figure 1 as the difference between the asymptote (total problems injected) and the regression line at the point of delivery (the dashed vertical line after TRR).

CRUD is an indicator of software quality. Since it is reasonable for small products to have less CRUD than large products, General Dynamics Decision Systems uses CRUD density (CRUD per KSLOC) as one of the software quality indicators.

Quality Results

Figure 2 examines the predicted quality improvement as projects progress up the various SEI CMM levels. This chart shows that predicted quality improves (which is synonymous with decreasing latent defect [CRUD] predictions) with increasing SEI CMM levels.

Quality Analysis

The metric data show that compared with an average Level 2 program, Level 3 programs have 3.6 times fewer latent defects, Level 4 programs have 14.5 times fewer latent defects, and Level 5 programs have 16.8 times fewer latent defects.

The improvement in quality is expected for projects that transition from Level 2 to 3 due to the Peer Review KPA found in Level 3. Peer reviews are widely recognized in the industry for being the single most important factor in detecting and preventing defects in software products. Quality is also expected to improve for projects transitioning from Level 3 to 4 due to the Quantitative Process Management and Software Quality Management KPAs. Using quality metric data such as peer review effectiveness and phase containment effectiveness will allow the project to modify its processes when the observed metric falls below the organizational and project control limits.

The improvement from Level 4 to Level 5 is attributable to the Defect Prevention and Process Change Management KPAs. Projects operating at this level perform Pareto analysis on the root cause of their problems and perform causal analysis to determine the process changes needed to prevent similar problems from occurring in the future.

It should be noted that large improvements in defect density are more readily obtained when the number of defects is large, as would be expected in lower maturity level projects. At higher maturity levels, it becomes more and more difficult to dramatically reduce the defect density.

2. Phase Containment Metric

Phase containment is defined as the ratio of problems detected divided by the number of problems inserted within a phase. For example, if 100 problems were introduced in detailed design but only 75 problems were detected by the peer-review process, then the phase containment effectiveness would be 75 percent. The General Dynamics Decision Systems' goal is at 85 percent phase containment effectiveness. Projects below this threshold perform causal analysis to improve their peer review and testing processes. The focus in improving phase containment is to catch problems as early as possible. The cost of fixing problems escalates dramatically the longer the problem remains undetected in the software life cycle.

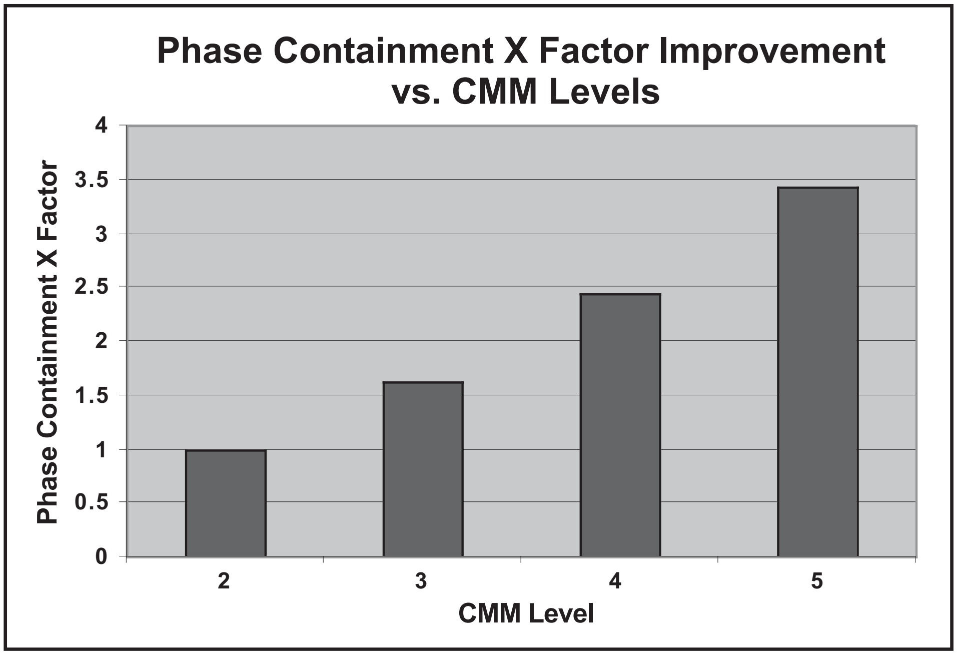


Figure 3: *Phase Containment Effectiveness Versus CMM Level*

Phase Containment Results

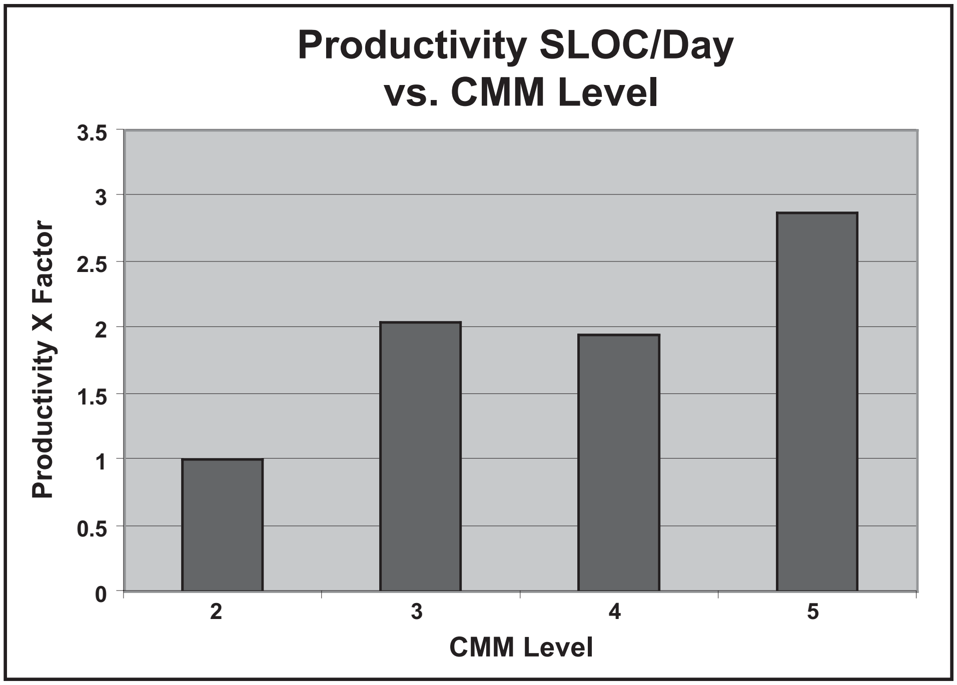
Figure 3 illustrates the Phase Containment Effectiveness improvements with respect to CMM level.

Phase Containment Analysis

Analysis of the data shows that compared with an average Level 2 program, Level 3 programs have 1.6 times better phase containment effectiveness, Level 4 programs have 2.4 times better phase containment effectiveness, and Level 5 programs have 3.4 times better phase containment effectiveness.

The improvement in phase containment effectiveness from Level 2 to 3 is primarily due to the Peer-Review KPA. Improvements from Level 3 to 4 are due to increased attention on peer review effectiveness using statistical process control charts and monitoring and removing assignable causes of variation, e.g., variation not inherent in the peer-review process. Improvements from Level 4 to Level 5 are attributable to the increased focus on the Defect

Figure 4: *Productivity Versus CMM Level*



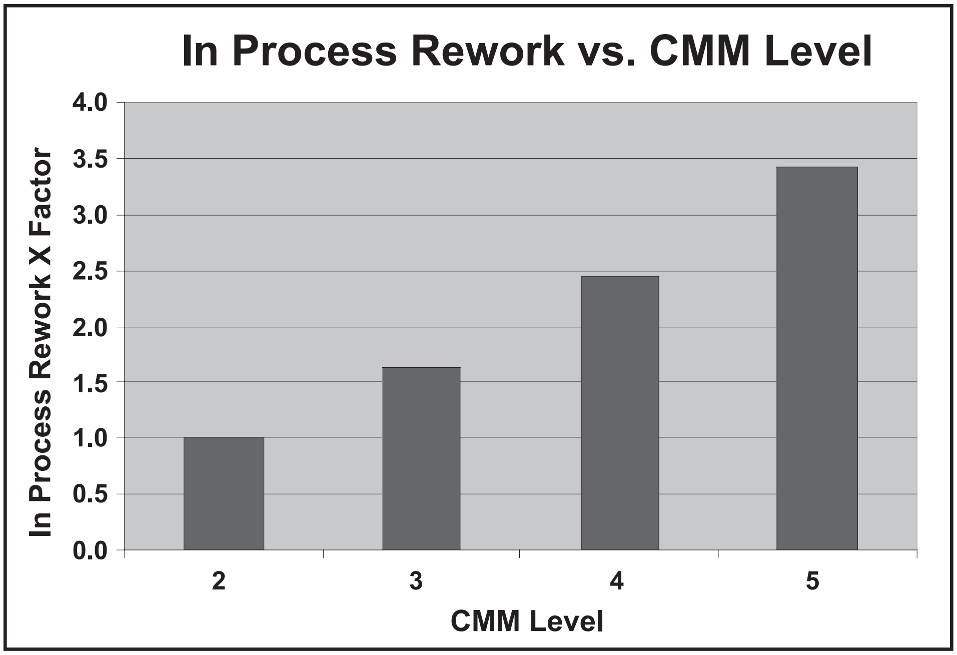


Figure 5: *In Process Rework Versus CMM Level*

Prevention and Process Change Management KPAs.

3. Productivity Metric

Productivity is defined as the amount of work produced divided by the time to produce that work. This may be measured in SLOC per hour, or some similar measure. Each project at General Dynamics Decision Systems tracks its productivity by measuring SLOC produced and the number of hours to produce that code.

Productivity Results

For proprietary reasons, the actual number of lines of code per hour is not shown; however, the relative productivity between projects at different levels of maturity can be seen in Figure 4 (see page 11). The data are normalized to the productivity of an average Level 2 project. The X factor is defined as the relative improvement as compared with a Level 2 program. For example, if a Level 2 program has an average productivity of eight SLOC per day and a Level 3 program has an average productivity of 16 SLOC per day, one could say that the Level 3 programs have a 2 X factor as compared with Level 2 programs.

Productivity Analysis

Project data show that compared with an average Level 2 program, Level 3 programs show a 2 X factor improvement in productivity, Level 4 programs show a 1.9 X factor improvement in productivity and Level 5 programs show a 2.9 X factor improvement in productivity.

Productivity is affected by factors other than process maturity, most importantly technology changes. For example, the data shown include projects that may have started before some form of automated code generation became available. In addition, the amount of code reuse on a project can greatly affect the productivity of that project. As projects increase their level of maturity, the ability to effectively reuse software source code is enhanced. Likewise, software code that is reused from a high maturity level project requires less rework and is more easily understood. These factors act as multipliers in the productivity of high maturity level projects.

It is interesting to note that in the transition from Level 3 to Level 4, projects do not experience a statistically significant change in productivity. This appears to be a side effect of Level 4 being a transitionary state in which projects quickly progress to Level 5 practices once making Level 4. Level 4 programs can monitor their critical processes using statistical process control techniques. However, the skills needed to perform causal analysis and process change are obtained at Level 5.

Table 2: *Model of Rework Costs per CMM Level for 100 KSLOC Program*

CMM Level	CRUD for 100 KSLOC	Post Release Rework (hrs)	Pre Release Rework (hrs)
Level 5	19.11567615	306	2,397
Level 4	22.0952381	354	3,358
Level 3	88.25757576	1,412	5,043
Level 2	315.8653846	5,054	8,208

It is also possible that the effective utilization of statistical process control and software quality management introduced at Level 4 becomes more effective over time and that benefits are not realized in the short term in this transitionary state. As with any new technology, it is expected that a cycle of absorption is needed before the full benefits can be observed.

4. Rework Metric

At General Dynamics Decision Systems, each software engineer enters his or her time card on a daily basis to include hours expended, charge number, and burden code. Burden codes measure major process activities and are subdivided into the following categories:

G - Generate
V - eValuate
K - reworK
S - Support

We measured the percentage of rework on a project as defined by "total hours for rework divided by total hours on the project." We then evaluated project results based upon CMM levels. Note that post-release rework due to maintenance is not included in this analysis.

Rework Results

The amount of rework is normalized to a Level 2 program where rework reduction is shown as an improvement in Figure 5. The X factor rework reduction is calculated by taking the percentage of rework for Level 2 programs and dividing the percentage of rework for all projects with a given CMM level.

As compared with an average Level 2 program, Level 3 programs show a 1.6 X factor reduction in rework, Level 4 programs show a 2.4 X factor reduction in rework, and Level 5 programs show a 3.4 X factor reduction in rework. It is interesting to note that the rework X factor improvements match the phase containment X factor improvements. This is not surprising due to the correlation of early in-process fault detection

CMM Level Transition	Cost for SPI in hrs (2.5% of Base)	Cost Savings on Rework (hrs)	Return on Investment
Level 4 to 5	884	1,009	14%
Level 3 to 4	1,310	2,744	109%
Level 2 to 3	2,544	6,806	167%

Table 3: *Return on Investment (ROI) by Level Transitions*

to the amount of reduced rework, i.e., better phase containment directly relates to lower rework.

Process Improvement Implementation Strategies

The following strategies are a result of several lessons learned from the software process improvements made at General Dynamics Decision Systems:

- Plan for organizational software process focus and definition impacts during reorganization planning.
- Statistical process control training and training on assignable causes of variation.
- Focus on new projects. It is extremely difficult to change projects, especially at a low maturity level, once they have started.
- A top down focus is essential before getting buried in the details of the CMM; start with the intent of each KPA and determine how it fits into your environment.
- Emphasize productivity, quality, and cycle time. Avoid process for the sake of process.
- Management commitment is needed from all levels; commitment from upper management won't be enough unless individual project leaders/managers are also determined to succeed.
- Practitioners and task leaders, not outside process experts, should be used to define processes.
- Managers need to be convinced of the value of process improvement; it's not free, but in the long run it certainly pays for itself.
- Copying process documents from other organizations usually does not work well; the process must match your organization.
- Overcoming resistance to change is probably the most difficult hurdle when climbing the CMM ladder.
- There are no silver bullets! Process change takes time, talent, and a commitment with which many organizations are uncomfortable. If it was easy, everyone would have already done it.

Return on Investment

The process improvement efforts to support 360 software engineers include the following:

- Full-time chief software engineer and metrics champion.
- Weekly software improvement meetings by software task leaders.
- Project kickoffs, phase-end reviews, and post mortems.
- Focused process improvement working groups by project personnel.

The above process improvement efforts were approximately 2.5 percent of the base staffing of 360 software engineers. Given the following assumptions for a single project:

- 100 KSLOC size project for two years.
- Sixteen hours to fix a defect found after software release.

The post and pre-release rework calculated from our return-on-investment (ROI) model is shown in Table 2.

Assuming 2.5 percent investment for process improvement on a project to progress one level of CMM maturity within one year, the ROI per CMM level is depicted in Table 3.

The ROI calculations do not take into account the added benefit of being able to apply existing resources for pursuit/execution of new business opportunities due to improved cycle times and earlier completion dates.

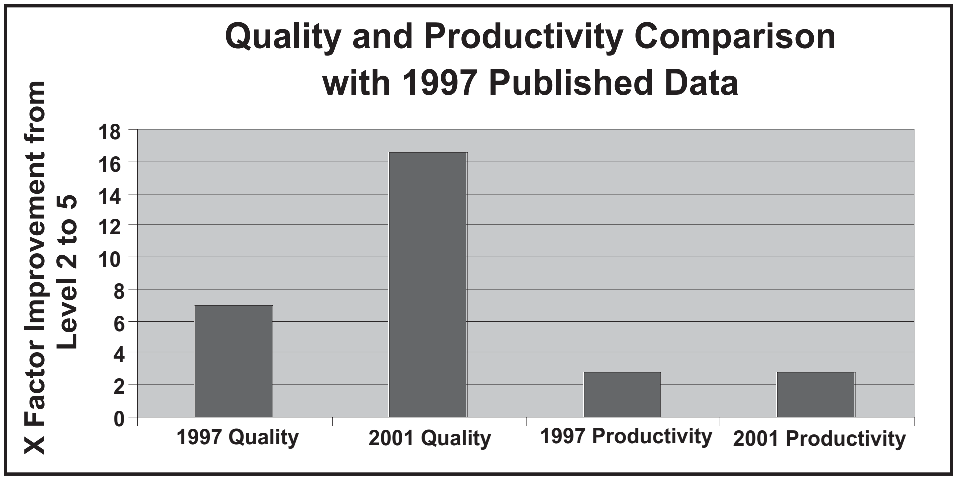
Conclusion

Similar to the earlier 1997 results published regarding General Dynamics Decision Systems' Software Process Improvement [3], each level of software maturity results in improved quality and productivity. Each level of CMM maturity reduces defect density by a factor of almost four on the average until Level 4 is reached, where a 16 percent improvement is seen from Level 4 to 5. Phase containment effectiveness and rework improve on the average by 50 percent with each maturity level. Productivity improves 100 percent for Level 2 to 3 transitions and by 50 percent for Level 4 to 5 transitions.

Comparing the General Dynamics Decision Systems' 2001 study with the published 1997 results shows some similarities and some differences as shown in Figure 6. The productivity improvement from Level 2 to 5 is about the same for both studies, around a 2.8 X factor improvement. The quality improvements, however, are more pronounced, 16.5 X factor improvement between Levels 2 and 5, than the General Dynamics Decision Systems' 1997 study that documented a 7 X factor improvement between Levels 2 and 5 [3]. This suggests that the quality benefits increase the longer an organization is able to maintain a Level 5 maturity capability.

The ROI analysis shows the largest benefit is going from Level 2 to 3 with 167 percent ROI. Level 3 to 4 advancement also shows a significant 109 percent ROI. Although the Level 4 to 5 ROI of 14 percent is not as significant as the other level transitions, subjective experience from these authors indicates that Level 4 projects are transitionary and short lived, quickly obtaining Level 5 much earlier than the one year per level

Figure 6: *Quality and Productivity Comparison with 1997 Published Data*



transition assumption used by the analysis model.

These data suggest that Level 5 is the most desirable state an organization would strive for in order to maximize the quality and productivity performance of a project.

Process improvement takes time to institutionalize and requires a commitment from management in order to succeed. Achieving higher levels of process maturity requires an investment of time and money in process improvements, including tool integration to aid in the collection and interpretation of quantitative data.

In conclusion, process improvement activities must be undertaken with a look at return on investment. Higher maturity organizations take this into account when initiating SPI activities. The CMM by itself does not assure improved performance results. Performance improvement must be specifically identified as the goal for SPI to avoid process for process sake. Tailoring of processes and a focus on cycle time are needed in addition to the traditional CMM emphasis.◆

References

1. Paulk, Mark, et al. "Capability Maturity Model Version 1.1." IEEE Software July 1993: 18-27.
2. Arkell, Frank, and Joseph Sligo, eds. "Software Problem Prediction and Quality Management." Conference Proceedings of the Seventh International Conference on Applications of Software Measure-ment, Oct. 1996.
3. Diaz, Michael, and Joseph Sligo, eds. "How Software Process Improve-ment Helped Motorola." IEEE Software Sept. 1997: 75-81.

About the Authors

Mike Diaz is a chief software engineer for the General Dynamics Decision Systems and is responsible for all aspects of software development in an organization of 360 software engineers. Diaz was a key contributor leading to General Dynamics Decision Systems' second Capability Maturity Model® Level 5 rating. Diaz's experience includes 19 years of software technical leadership in requirements management, systems engineering, security architectures, and secure key management systems while at General Dynamics Decision Systems. Diaz has been awarded membership in General Dynamics Decision Systems' Scientific Advisory Board Association, the highest technical association within General Dynamics Decision Systems. Diaz received a bachelor's of science degree in electrical engineering and a master's degree in computer engineering from Boston University.

General Dynamics Decision Systems
8220 E. Roosevelt St.
Scottsdale, AZ 85252
Fax: (480) 675-2398
E-mail: michaeldiaz@gd-decision systems.com

Jeff King is a senior recruiter for General Dynamics Decision Systems in Scottsdale, Ariz. He has worked for General Dynamics Decision Systems for six years and has 15 years total experience in technical recruiting. During his career, King has supported information technology, telecommunications, aerospace, and defense industries. He received a bachelor's degree in economics from Northern Arizona University in Flagstaff, Ariz. in 1979 and earned 60 hours post graduate at Arizona State University.

General Dynamics Decision Systems
8220 E. Roosevelt St.
Scottsdale, AZ 85252
Phone: (480) 675-2995
E-mail: jeff.king@gd-decision systems.com

Coming Events

March 25-28
Software Test
Automation Conference
San Jose, CA
www.sqe.com/testautomation

April 8-10
Secure E-Business Executive Summit
SecurE-Biz.net
Arlington, VA
www.secure-biz.net

April 9-10
Southeastern Software
Engineering Conference
Huntsville, AL
www.ndia-tvc.org/SESEC2002/

April 29–May 2
Software Technology Conference 2002
"Forging the Future of Defense
Through Technology"
STC
Salt Lake City, UT
www.stc-online.org

May 13-17
Software Testing
Analysis and Review
(STAREAST 2002)
STAR EAST 2002
Orlando, FL
www.sqe.com/stareast

June 3-6
Combat Identification Systems
Conference
Colorado Springs, CO
www.usasymposium.com

July 22-25
Joint Advanced Weapons Systems Sensors,
Simulation, and Support Symposium
(JAWS S3)
Colorado Springs, CO
www.jawswg.hill.af.mil