feature

process improvement

Are CMM Program Investments Beneficial? Analyzing Past Studies

Daniel Galin, *Ruppin Academic Center*

Motti Avrahami, *Lipman Electronic Engineering*

A review of the literature on CMM program investments indicates that they improved performance consistently for seven common software development metrics.

CMM experts strongly believe that investments in programs promoting an organization's CMM maturity yield substantial organizational and economic benefits. In particular, they argue that CMM programs that implement software process improvements can provide these benefits:

- reduced error density in developed software,
- fewer required regression tests and correction cycles,
- fewer software errors detected after software release, and
- more project milestones completed on time (improved schedule fidelity).

Moreover, they claim that these benefits lead to substantial economic gains, such as

- reduced development costs due to increased productivity, better development methods, improved design review methods, and so on;
- substantially reduced costs for testing and correcting software; and
- reduced compensation paid to customers for faulty software or late completion.

The published results of such CMM programs cover a wide variety of industries, performance metrics, and CMM levels. These results' scattered nature makes the arguments for the programs' effectiveness less convincing. So, we sought to provide more solid evidence through a combined analysis of past studies' results. Our bibliographic survey focused on publications that presented quantitative, empirical results for CMM advancements (measured by *CMM level transition*, or CMMLT). We believe that our technique will work well for combined analysis of empirical data from a great variety of sources.

Planning the analysis

We chose 19 papers, which reported on results for more than 400 projects from organizations in several countries.[1–19] (For information on related papers that we didn't choose, see the "Related Work in CMM and Process Improvement" sidebar.) Each paper described an organization's experiences implementing CMM over a two- to 10-year period. No one paper provided conclusive evidence of the benefit of CMM program investment. However, we felt that combining and analyzing the papers' results would produce the required evidence.

0740-7459/06/$20.00 © 2006 IEEE

Related Work on CMM and Process Improvement

Brad Clark comprehensively investigated the relationship between the developer's CMM level and software development efficiency (productivity) for 112 projects.[1,2] He concluded that an increment of one CMM level reduced development effort by 15 to 21 percent (and increased productivity by 18 to 26 percent).

The second phase of Clark's research examined 161 projects. In this phase, an increment of one CMM level decreased development effort 4 to 11 percent (and increased productivity 4 to 12 percent). One reason for these substantially lower results might be that they weren't based on a direct measurement of productivity changes in the same organization. Instead, they were based on projects that were from different organizations and that might have differed in nature and used different development tools and environments. Another factor might be that the multivariate model Clark used didn't fully represent the deficiencies of using KLOC as a measure for project size.

Donald Harter, Mayuram Krishnan, and Sandra Slaughter analyzed 30 large-scale projects at a major IT firm over 12 years. They found that an increase in CMM level was associated with a statistically significant increase of development productivity, increased software quality, and a shorter development time cycle.[3] In an extensive review of publications on the implementation of process methods, including CMM, Khaled El Emam and Dennis Goldenson reported qualitative performance improvements in terms of higher productivity, higher quality, increased ability to meet schedules, and higher customer satisfaction.[4] On the basis of an investigation of 31 projects, Patricia Lawlis, Robert Flowe, and James Thordahl reported improved cost and schedule performance as the CMM level increased.[5]

Goldenson and Diane Gibson presented findings for 12 diverse case studies.[6] The results, which were mostly quantitative and were specific for each organization, referred to costs, schedule, quality, customer satisfaction, and ROI. For example, for cost, an increase in CMM level produced

- 33 percent decrease in cost to fix an error (Boeing, Australia),
- 20 percent reduction in unit software costs (Lockheed Martin Management and Data Systems), and
- 30 percent improvement in software development productivity (Lockheed Martin M&DS).

This study revealed some of the possible benefits of an increase in CMM level. However, it didn't indicate the CMM level of most of the organizations or the *CMM level transition* due to investments in CMM programs. (A CMMLT is a transition from one CMM level to another.) Peter Capell published a more recent comparative study, which presented separate results for 11 defense organizations.[7] Similarly to Goldenson and Gibson's study, Capell's findings indicate performance benefits for all the projects, where specific and no common metrics were used. The reported results didn't suit the format needed for our study.

For our analysis, we needed to define a normalization scheme that would let us draw conclusive results from data on CMM improvement at widely varying organizations. For example, the 19 papers reported on organizations of varying sizes that used different coding standards and development environments. The papers also described different development tasks, performance metrics, and CMM levels.

So, in each paper we searched for the percentage of performance improvement related to a defined increase in CMM level. We created a database that contained the following data from each paper:

- the initial CMM level,
- the final CMM level,
- the initial value of a performance metric, and
- the final value of the performance metric.

On the basis of this data, we were able to define a CMMLT performance improvement record (or *CMMLT record*) that indicated the CMMLT, the performance metric, and the percentage of performance improvement. A paper might yield more than one CMMLT record when it deals with more than one performance metric or CMMLT. These percentage of performance improvement results (based on "after-before ratio" results) are practically independent of the studied organization's measurement methods.

From the 19 papers, we created 99 CMMLT records:

- 85 single CMMLTs,
- 11 double CMMLTs,
- two triple CMMLTs, and
- one quadruple CMMLT.

(Single CMMLTs cover an increase of one CMM level; the others cover increases of multiple levels.) The 99 records covered seven common performance metrics:

In addition, George Issac, Chandrasekharan Rajendran, and R.N. Anantharaman recently compared how CMM level affected the performance of noncertified organizations, medium-maturity organizations (ISO 9000 certified and CMM Level 3), and high-maturity organizations (CMM Levels 4 and 5).[8,9] The researchers distributed a questionnaire to 1,200 software development professionals in 100 organizations, 324 of whom completed it. The questionnaire asked the respondents to rate their organization on a seven-point scale on matters pertaining to quality and ROI.

Most relevant to our subject is the comparison of medium-maturity organizations to high-maturity organizations. The researchers processed 73 responses for the former and 188 responses for the latter. The mean quality rating was 4.55 for medium-maturity organizations and 5.27 for high-maturity organizations (the highest-possible score being 7). The mean ROI rating was 4.60 for medium-maturity organizations and 5.02 for high-maturity organizations. The differences in quality and ROI ratings between the two maturity levels were highly statistically significant: 0.004 significance for quality and 0.000 significance for ROI.

However, these results have two limitations. First, because they're based on only 324 responses out of 1,200 distributed questionnaires, some biases are possible. Second, they're based on qualitative evaluations, so they aren't useful for economic evaluations.

Because the methodological and data-collection-processing methods in all these papers produced data that differed from our required format, we couldn't use that data for our study. However, the results do indicate the expected benefits of transfer to higher CMM levels.

References

1. B.K. Clark, "Quantifying the Effects of Process Improvement on Effort," *IEEE Software*, vol. 17, no. 6, 2000, pp. 65–70.
2. B.K. Clark, "Effects of Process Maturity on Development Efforts," doctoral dissertation, Center for Software Eng., Univ. of Southern California, 1997.
3. D.E. Harter, M.S. Krishnan, and S.A. Slaughter, "Effects of Process Maturity on Quality, Cycle Time, and Effort in Software Product Development," *Management Science*, vol. 46, no. 4, 2000, pp. 451–466.
4. K. El Emam and D.R. Goldenson, "An Empirical Review of Software Process Assessments." *Advances in Computers*, vol. 53, 2000, pp. 319–422.
5. P.K. Lawlis, R.M. Flowe, and J.B. Thordahl, "A Correlational Study of the CMM and Software Development Performance," *Crosstalk*, Sept. 1995; www.stsc.hill.af.mil/crosstalk/frames.asp?uri=1995/09/Correlat.asp.
6. D.R. Goldenson and D.L. Gibson, *Demonstrating the Impact and Benefits of CMMI: An Update and Preliminary Results*, tech. report CMU/SEI-2003-009, Software Eng. Inst., Carnegie Melon Univ., 2003.
7. P. Capell, *Benefits of Improvement Efforts*, tech. report CMU/SEI-2004-SR-010, Software Eng. Inst., Carnegie Mellon Univ., 2004.
8. G. Isaac, C. Rajendran, and R.N. Anantharaman, "Do Quality Certifications Improve Software Industry's Operational Performance?" *Software Quality Professional*, vol. 6, no. 1, 2003, pp. 30–37.
9. G. Isaac, C. Rajendran, and R.N. Anantharaman, "Do Quality Certifications Improve Software Industry's Operational Performance?—Supplemental Material," *Software Quality Professional*, vol. 6, no. 1, 2003; www.asq.org/pub/sqp/past/vol6_issue1/issac.pdf.

- error density,
- software development productivity,
- percentage of rework,
- cycle time for the completion of a typical software project,
- schedule fidelity,
- error detection effectiveness, and
- return on investment (ROI).

Table 1 classifies the 19 publications according to these metrics.

After creating the records, we sorted them according to performance metric and CMMLT.

While searching for appropriate papers to analyze, we also found 12 CMMI (CMM Integration) papers, most of which dealt with the CMM-to-CMMI transition. The CMM–CMMI papers reported quantitative data on improved quality due to that transition. However, the low number of CMMI empirical records in the 12 papers made them insufficient for our secondary analysis. For a list of the 12 papers, see www.ruppin.ac.il/Galin/CMMI.

The results

Table 2 presents the median improvement and range of improvement for all 99 CMMLT records. We focus here on the findings for the 85 single CMMLT records. (Because of the small number of records for the other CMMLTs, we couldn't use them for statistical analysis.)

Error density

The reporting organizations measured error density as the number of errors per KLOC. Nineteen CMMLT records dealt with this metric. The median improvement for a single CMMLT ranged from 26 to 63 percent.

Productivity

Most reporting organizations applied the classic KLOC/SM (thousands of lines of code per staff month) productivity measure. Sixteen

Table 1

CMM secondary analysis sources

Researcher	Performance metric: Error density	Productivity	Rework	Cycle time	Schedule fidelity	Error detection effectiveness	Return on investment
McGarry[1]	X	X		X			
King[2]	X	X	X			X	X
Haley[3]	X	X	X				X
Pitterman[4]	X			X	X		X
Diaz[5]	X	X		X			X
McGarry[6]	X	X					X
Yamamura[7,8] and Vu[9]*	X	X	X	X		X	X
Wohlwend[10]	X			X	X	X	X
Ferguson[11]	X	X	X	X	X		X
Oldham[12]				X			X
Keeni[13]			X	X			
Lowe[14]				X			X
Herbsleb[15]		X		X		X	
Reo[16]		X					
Frazer[17]	X						
Curtis[18]	X			X	X		
Blair[19]	X	X		X			X

All three articles relate to the same empirical observations at the Boeing Defense and Space Group.

single CMMLT records dealt with this metric; median improvement ranged from 26 to 187 percent.

Rework

The organizations used metrics relating to rework time during development and corrective maintenance time as a percentage of the total project development resources. Ten single CMMLT records dealt with rework; median improvement ranged from 34 to 40 percent. (Some organizations reported that error detection improvements somewhat increased rework during development but substantially decreased it during maintenance.)

Cycle time

Improvements in cycle time refer to project duration becoming shorter. The improvement is estimated by comparing the time to complete projects of similar size and nature. Thirteen single CMMLT records dealt with this metric; median improvement ranged from 28 to 53 percent.

Schedule fidelity

Most reporting organizations measured this metric as the percentage of projects and milestones completed on or before the scheduled time. Five single CMMLT records dealt with this metric, for the two lowest CMMLTs (CMM 1 to CMM 2 and CMM 2 to CMM 3). These CMMLTs showed substantial median improvement: 37 and 46 percent.

Another result relates to a double CMMLT (CMM 3 to CMM 5) with a marginal 6 percent improvement. Does this poor result signify that lower-level CMMLTs (as in the previous paragraph) are more likely to show major improvements, or is it a singular event that justifies no conclusions?

Error detection effectiveness

Most organizations defined error detection effectiveness (or error-screening effectiveness) as the percentage of software errors detected during development out of the total project errors (including those detected during the software's operational stage). Thirteen single

Table 2

Performance improvement according to metric and CMM transition level*

	Metric						
CMMLT	Error density	Productivity	Rework	Cycle time	Schedule fidelity	Error detection effectiveness	ROI
CMM 1 to CMM 2							
Median improvement	29	26	38	35	46	70	650
Range of improvement	16–75	25–74	31–51	26–69	33–79	60–80	110–900
No. of results	6	3	3	4	3	2	4
CMM 2 to CMM 3							
Median	60	42	40	28	37	70	225
Range	28–77	9–100	38–50	—	34–40	42–93	180–270
No. of results	9	10	3	6	2	7	2
CMM 3 to CMM 4							
Median	63	187	38	46	—	74	210
Range	50–75	—	34–42	—	—	—	—
No. of results	2	1	2	1	—	1	1
CMM 4 to CMM 5							
Median	26	37	34	53	—	13	440
Range	14–37	22–52	28–40	36–70	—	10–14	110–770
No. of results	2	2	2	2	—	3	2
CMM 1 to CMM 3							
Median	95†	91†	65	83	—	—	790
Range	—	12–170	—	—	—	—	—
No. of results	1	2	1	1	—	—	1
CMM 1 to CMM 5							
Median	—	—	—	—	—	—	1,900
Range	—	—	—	—	—	—	—
No. of results	—	—	—	—	—	—	1
CMM 2 to CMM 4							
Median	—	150	—	23	—	—	—
Range	—	—	—	—	—	—	—
No. of results	—	1	—	1	—	—	—
CMM 2 to CMM 5							
Median	—	—	—	—	—	—	395
Range	—	—	—	—	—	—	110–670
No. of results	—	—	—	—	—	—	2
CMM 3 to CMM 5							
Median	104	47	—	—	6	—	—
Range	—	—	—	—	—	—	—
No. of results	1	1	—	—	1	—	—

** The top half of the table comprises single CMMLTs; the bottom half comprises multiple CMMLTs. For median improvement and range of improvement, results are in percentages.*

† These double CMMLT results represent a case in which the organization implemented a double-transition policy, where the first transition focused on error density and the second focused on productivity.

CMMLT records dealt with this metric, each showing substantial improvement. While the three lowest CMMLTs showed relatively high median improvement (70 to 74 percent), the highest CMMLT (CMM 4 to CMM 5) showed relatively modest improvement (13 percent). These results were similar to those for schedule fidelity.

ROI

The reporting organizations applied various methodologies to measure their ROI. The varying conditions and methodologies yielded a wide range of ROI results for the different CMMLTs, from 120 to 650 percent.

Nine single CMMLT records dealt with this metric. All nine showed positive ROI; in other

Table 3

T-test results for the significance of performance improvement

Metric	No. of results	Mean improvement performance (%)	Standard deviation (%)	Calculated t	$t_{0.01}$*	Null hypothesis verified or rejected
Error density	19	48	22.3	9.4	2.55	Rejected
Productivity	16	52	43.6	4.8	2.60	Rejected
Percentage of rework	10	39	7.4	16.8	2.82	Rejected
Cycle time	13	38	16.9	8.0	2.68	Rejected
Schedule fidelity	5	45	19.0	5.5	3.75	Rejected
Error detection effectiveness	13	63	24.6	7.7	2.68	Rejected
Return on investment	9	360	330.0	3.9	2.96	Rejected

*$t_{0.01}$ is a value of the t distribution that determines whether the hypothesis will be rejected.

words, they provided economic incentive for the organizations to continue investing in software process improvement. The median ROI was 360 percent.

Verifying the results

To provide additional support for our findings, we performed statistical analysis of the CMMLT records. We assumed that the law of large numbers applies in our case and that our records were samples out of a normal distribution. Accordingly, we carried out a t-test for each performance metric. We wanted to test this null hypothesis:

> *The mean performance improvement reported for a single CMMLT is zero.*

The t-test results rejected the null hypothesis for all seven metrics, with high significance ($p < 0.01$). Table 3 shows the detailed results.

The wide range of improvement results for some of the single CMMLTS for a performance metric reflects the diversity between organizations that we mentioned before. However, the median improvement and range of improvement exhibit a consistent, positive picture for each performance metric and for each single CMMLT.

However, some readers might claim that researchers prefer to publish success stories and avoid failure stories. They might also claim that the fact that all the published results are positive and that none mention failure might serve as proof of the bias in the available data. However, they should take into account that most of the published data is actually averages of many projects, some of which could be failure cases.

Our database's size (more than 400 projects from 19 information sources) leads us to accept our secondary analysis results, although with some reservations regarding possible data bias. We believe that the actual bias is small, doesn't really affect our analysis, and doesn't change our main conclusion: that investment in CMM programs leads to improved software development and maintenance.

Acknowledgments

We gratefully acknowledge Stan Rifkin, *IEEE Software* associate editor in chief, for his helpful comments on earlier versions of the article. An earlier version of the article was presented at the 2005 IEEE International Conference on Software—Science, Technology, and Engineering (SwSTE 05).

References

1. F. McGarry et al., *Software Process Improvement in the NASA Software Engineering Laboratory*, tech. report CMU/SEI-94-TR-22, Software Eng. Inst., Carnegie Mellon Univ., 1999.
2. J. King and M. Diaz, "How CMM Impacts Quality, Productivity, Rework, and the Bottom Line," *Crosstalk*, Mar. 2002, pp. 9–14; www.stsc.hill.af.mil/crosstalk/2002/03/diaz.html.
3. T. Haley et al., *Raytheon Electronic Systems Experience in Software Process Improvement*, tech. report CMU/SEI 95-TR-17, Software Eng. Inst., Carnegie Mellon Univ., 1995.
4. B. Pitterman, "Telcordia Technologies: The Journey to High Maturity," *IEEE Software*, vol. 17, no. 4, 2000, pp. 89–96.
5. M. Diaz and J. Sligo, "How Software Process Improvement Helped Motorola," *IEEE Software*, vol. 14, no. 5, 1997, pp. 75–81.
6. F. McGarry and B. Decker, "Attaining Level 5 in CMM Process Maturity," *IEEE Software*, Nov./Dec. 2002, pp. 87–96.
7. G. Yamamura and G.B. Wigle, "SEI CMM Level 5: For

the Right Reasons," *Crosstalk*, vol. 10, no. 8, 1997, pp. 3–6.
8. G. Yamamura and G. B. Wigle, "Practices of an SEI CMM Level 5 SEPG," *Crosstalk*, vol. 10, no. 11, 1997, pp. 19–22.
9. J.D. Vu, "Software Process Improvement Journey (From Level 1 to Level 5)," keynote address presented at the 2nd Ann. European Software Eng. Process Group Conf., 1997; www.processgroup.com/john-vu-keynote2001.pdf.
10. H. Wohlwend and S. Rosenbaum, "Software Improvements in an International Company," *Proc. 15th Int'l Conf. Software Eng.* (ICSE 93), IEEE CS Press, 1994, pp. 212–220.
11. P. Ferguson et al., *Software Process Improvement Works!* tech report CMU/SEI-99-TR-027, Software Eng. Inst., Carnegie Mellon Univ., 1999.
12. L.G. Oldham et al., "Benefits Realized from Climbing the CMM Ladder," *Crosstalk*, vol. 12, no. 5, 1999, pp. 7–10.
13. G. Keeni, "The Evolution of Quality Processes at Tata Consultancy Services," *IEEE Software*, vol. 17, no. 4, 2000, pp. 79–88.
14. D.E. Lowe and G.M. Cox, "Implementing the Capability Maturity Model for Software Development," *Hewlett-Packard J.*, Aug. 1996, pp. 1–11.
15. J. Herbsleb et al., *Benefits of CMM-Based Software Process Improvement: Initial Results*, tech. report CMU/SEI-94-TR-013, Software Eng. Inst., Carnegie Mellon Univ., 1994.
16. D. Reo, "CMM for the Right Reasons," presentation at ASQF CMM Day, European Software Inst., 9 Apr. 2002.
17. K. Frazer, "Return on Software Process Improvement," 1997, pp. 15–16; home.earthlink.net/~krfrazer/Process_ROI.pdf.
18. B. Curtis, "The Cascading Benefits of Software Process Improvement," presentation at the 2nd Int'l Conf. Product Focused Software Process Improvement (PROFES 00), 2000.
19. R.B. Blair, "Software Process Improvement: What Is the Cost? What Is the Return on Investment?" presentation at Pittsburgh Project Management Inst. Chapter meeting, 12 Apr. 2001; www.pittsburghpmi.org/documents/meetings/presentations/PmiCmmSpiPresentation.ppt.

About the Authors

Daniel Galin is the head of information systems studies at the Ruppin Academic Center and an adjunct senior teaching fellow at the Technion's Faculty of Computer Science. His main research interests include software quality assurance and analysis and the design of information systems. He received his DSc in operations research from the Technion. He's a member of the IEEE Computer Society, the Information Technology Association of Israel, and the Israel Society for Quality. He recently authored *Software Quality Assurance—Theory to Implementation* (Addison Wesley, 2004). Contact him at the Ruppin Academic Center, Emek Hefer, Israel, 40250; dgalin@bezeqint.net.

Motti Avrahami is the operations quality control manager at Lipman Electronic Engineering (owned by VeriFone since November 2006). He has more than eight years' experience in software quality process and software testing. He received his MSc in quality assurance and reliability from the Technion. Contact him at Harchava 138, Moshav Mazor 73160, Israel; mottia@gmail.com.

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