



Measurement and Analysis in Capability Maturity Model Integration Models and Software Process Improvement¹

Dennis R. Goldenson
Software Engineering Institute

Joe Jarzombek
Office of the Under Secretary of Defense (Acquisition, Technology and Logistics)

Terry Rout
Griffith University

The explicit incorporation of measurement and analysis as a distinct process area in the Capability Maturity Model Integration (CMMI[®]) provides management with the visibility and focus that organizations need to guide the use of measurement in their process improvement efforts, which was missing in previous models. This article reviews the content and rationale behind the new process area and describes how the ideas introduced there are further elaborated and evolved throughout capability maturity model integration models.

Measurement, or the need for it, is pervasive in software and systems engineering. Yet an understanding of how to best use measurement has remained all too uncommon, as has straightforward guidance from the experts.

Even experts have difficulty following the sometimes-implicit threads among the measurement-related concepts in many process improvement models and standards. Exhortations to do measurement, without sufficiently elaborating what to do, have not worked exceptionally well in the past. However, there has been an increasing recognition of the importance of focusing explicitly on measurement and analysis. Certain basic ideas must be introduced early and well.

The Need for an Early Focus

The need to focus on measurement and analysis from the beginning of a process improvement effort has not always been well understood, even by experienced assessors and expert consultants. Yet organizations that have succeeded in putting successful measurement programs in place often say they could have avoided much grief and struggle with rework had they focused on how to implement measurement and analysis correctly in the earlier phases of their process improvement efforts.

Indeed, some organizations have created their own measurement process areas to guide their improvement efforts, become more competitive, and enhance their ability to more quickly achieve a higher level of process maturity. In fact, measurement is so important to the success of software projects that the U.S. Department of Defense requires the following of all major programs:

... to have a software measurement process to plan and track the software program and to assess and improve the development

process and associated software product. [1]

Focusing on measurement can provide much value to both projects and organizations. Measurement enables project managers to answer the following questions:

- Is there really a problem?
- How big is the problem?
- What is the scope of the problem?

“Measurement helps provide objective insight into issues and processes, along with the ability to objectively identify and manage risks and provide early detection and resolution of problems.”

- What is causing the problem?
- Are there related problems?
- Can I trust the data?
- What should I expect; what will happen?
- What are my alternatives?
- What is the recommended course of action?
- When can I expect to see results?

Measurement helps provide objective insight into issues and processes, along with the ability to objectively identify and manage risks and provide early detection and resolution of problems. Measurement also facilitates evidence-based team communication and enables objective planning and estimating and the ability to assess organizational performance in an unbi-

ased and defensible manner. These in turn provide an objective basis for defending and justifying decisions. Finally, measurement provides information that improves decision making in time to affect the business or mission outcome [2].

Doing Measurement Right

Of course, many approaches to software measurement now exist. Several published international standards address software measurement and closely related issues. There also exists voluminous literature in software measurement and metrics, and a rich body of literature in statistics and quantitative methods dating back well over a century.

Negotiating the morass of standards, models, guidebooks, courses, and expert consultants can be a daunting task. Fortunately, though, there is a clearly emerging community of practice that spans both software measurement and process improvement². In fact, the Capability Maturity Model Integration (CMMI[®]) Measurement and Analysis process area was developed in a collaborative, coordinated fashion with colleagues who also worked concurrently with the Practical Software and Systems Measurement Support Center, and worked on the development of the emerging ISO [International Organization for Standardization] standards on both software measurement and process assessment. People working in the field are also closely coupled with related work in standards and with groups such as the International Function Point Users Group (IFPUG)³.

Measurement and Analysis in Capability Maturity Model Integration Models

The Measurement and Analysis process area is an important addition to the CMMI. Its scope is much wider and more explicit than the treatment of measure-

ment in the Capability Maturity Model for Software (SW-CMM) [3]. The SW-CMM contains a measurement and analysis common feature, the practices of which apply to the institutionalization of the model's key process areas. Akin to generic practices in capability maturity model integration models, these practices are meant to control and improve the performance of the processes themselves. Some measurement-related practices also exist in various places in the activities-performed common feature in the SW-CMM, but a single, coherent treatment does not exist for what is required to establish and sustain a viable measurement and analysis process.

"The purpose of measurement and analysis is to develop and sustain a measurement capability that is used to support management information needs" [4]. The Measurement and Analysis process area supports all process areas by providing practices that guide projects and organizations in aligning their measurement needs and objectives with a measurement approach that will provide objective results that can be used in making informed decisions and by taking appropriate corrective actions.

As discussed more fully in the process area itself, measurement and analysis practices are organized under two specific goals that are aimed at (1) aligning measurement activities with identified information needs and objectives, and (2) providing data analyses and results that address those needs and objectives. These goals may be achieved by the successful performance of their respective specific practices shown in Figure 1⁴.

The specific practices associated with the first goal establish a coherent plan for measurement and analysis. They address these questions: "Why are we measuring?" "What are we going to measure?" "How are we going to measure?" and "What will be done with the data once we have them?" The specific practices associated with the second goal advise the user to just do it. Of course, the ultimate goal is as follows:

... to get the results of performing measurement and analysis into the hands of those who will take action based on the results. The process area emphasizes the need that results must be communicated to those needing the information. [5]

Other guidance about what constitutes good measurement practices does exist in capability maturity model integration

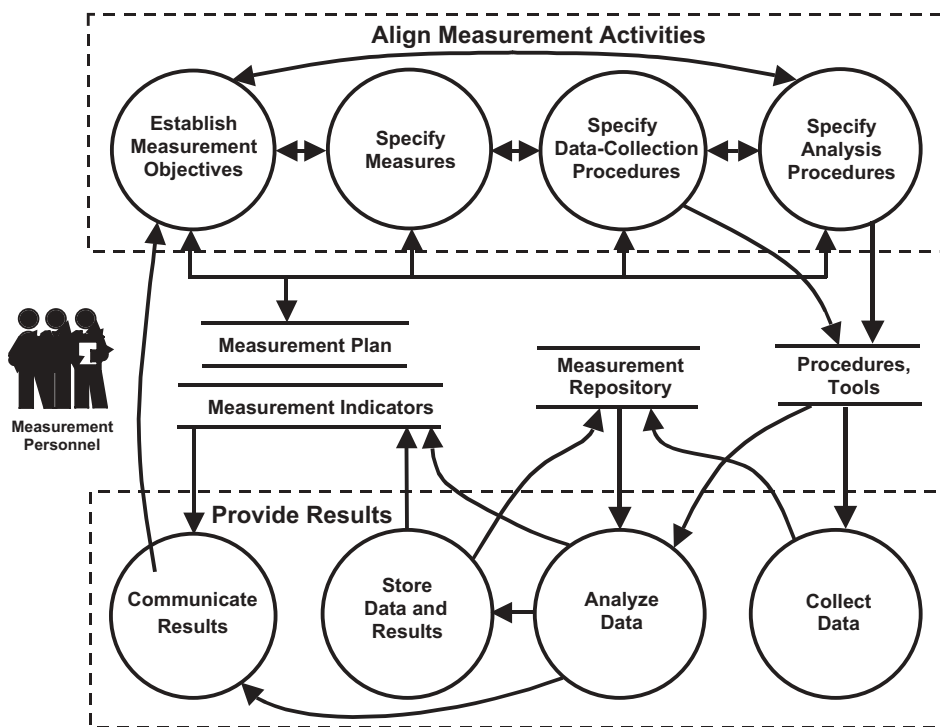


Figure 1: The Measurement and Analysis Process Area⁵

models, most notably in some of the process areas with a legacy in the source documents on which the CMMI is based. Those process areas do contain certain specific practices that require measurement activities to be performed. As seen more fully in the CMMI, the Measurement and Analysis process area makes explicit reference to other process areas – in particular, Organizational Process Definition and the heavily measurement-oriented process areas at CMMI Levels 4 and 5. With the addition of the Measurement and Analysis process area, the CMMI summarizes much of the experience base on which the proper conduct of measurement and analysis relies.

Maturing Measurement Capability

The Measurement and Analysis process area provides a central focus that describes good measurement practice. But the process area does not stand alone. The

CMMI also provides important guidance in its generic goals and practices, some of which have explicit measurement content. The generic practices serve together to help institutionalize measurement and analysis, or any other process, and to improve the capability with which measurement and analysis are performed over the life cycle of the product and organization.

Like any other process area, measurement and analysis can progress from being performed in an essentially ad hoc manner, through following a well-defined measurement process, to using measurement to evaluate and improve the measurement process itself. Several CMMI generic practices have a clear measurement flavor (Table 1). However, all of the generic practices can be applied to the conduct of measurement and analysis⁶.

Several generic practices discuss organizational policies, sufficiency of resources, explicit assignment of responsibilities, and training provisions. These help estab-

Table 1: Measurement-Related Generic Practices [4]

Practice	Statement
2.8 Monitor and Control the Process	Monitor and control the process against the plan for performing the process and take appropriate corrective action.
3.2 Collect Improvement Information	Collect work products, measures, measurement results, and improvement information derived from planning and performing the process to support the future use and improvement of the organization's processes and process assets.
4.1 Establish Quantitative Objectives for the Process	Establish and maintain quantitative objectives for the process about quality and process performance based on customer needs and business objectives.
4.2 Stabilize Sub-Process Performance	Stabilize the performance of one or more sub-processes of the process to determine its ability to achieve the established quantitative quality and process performance objectives.
5.1 Ensure Continuous Process Improvement	Ensure continuous improvement of the process in fulfilling the relevant business goals of the organization.
5.2 Correct Root Causes of Problems	Identify and correct the root causes of defects and other problems in the process.

lish and sustain process capability and a commitment to doing measurement regularly and well. Indeed, they provide the organizational infrastructure that is necessary to implement and institutionalize any process.

Other generic practices provide guidance for planning and related activities. The planning-related generic practices, including the establishment of quantitative objectives (4.1) and improvement objectives (5.1) help establish the scope and objectives for measurement work. Although they relate to both specific goals of the Measurement and Analysis process area, they are particularly important for the alignment activities discussed in specific goal 1 listed earlier.

Several generic practices, including stabilize sub-process performance (4.2), guide the performance and management of any process. These also support both specific goals 1 and 2 of Measurement and Analysis, but are particularly important for doing measurement and analysis and reporting the results.

Finally, along with others, the measurement-oriented generic practices that require monitoring and controlling the process (2.8), collecting improvement information (3.2), and correcting root causes of problems (5.2) help evaluate and improve the conduct of the measurement process itself. Together they provide an evidential basis for improving the manner in which future measurement and analysis are done⁷.

Implementing Good Measurement Practice

Along with the measurement-related generic practices, the Measurement and Analysis process area provides essential guidance about what to do whenever there is a need for measurement. However, measurement and analysis are always done in the context of performing other processes.

The Measurement and Analysis process area seamlessly integrates measurement activities with those other processes to address a wide variety of both project and organization-wide information needs and provides a basis for integrating measurement and analysis with process definition.

Like other support functions, the Measurement and Analysis process area serves multiple purposes. Every process area is dependent to some extent on properly using measurement and analysis. The engineering and management process areas describe the sources of the

contractual requirements, other information needs, and business objectives with which the measurement and analysis activities are aligned. In turn, the results of the measurement activities are provided back to inform the work described by those same process areas.

Maturing Analytic Capability

Measurement is applied differently as the organization successfully satisfies the goals of more and more CMMI process areas. It typically begins with a focus on clarifying sometimes-implicit business objectives and informational needs and translating them into measurable objectives. A basic set of skills, resources, and

“Moreover, relying on a well-defined measurement process can demonstrate evidence of business value, thus helping justify continued investment in process improvements and the measurement and analysis activities that support it.”

experiences is built for the future. Measurement often starts with using simple charts and graphs, but as the organization matures, demand increases for more sophisticated quantitative analyses such as statistical process control (SPC), structural modeling, or other multivariate statistical methods.

As the organization's analytic sophistication increases, finer-grained measures of defects and product quality are coupled explicitly with process performance. By the time the organization reaches CMMI Level 4, routine reliance on quantitative management enhances process discipline. After Level 5 is attained, there is increased use of work-product inspections, systematic programs of defect prevention driven by causal analysis, and improved process performance that often leads to markedly increased predictability of productivity and schedule.

Analytic Approaches

Many statistical analytic solutions to measurement problems are possible in software process improvement; however, few are widely used. For example, in an unpublished survey of representatives of high-maturity organizations, almost 90 percent reported that SPC control-charting techniques were in common or standardized use in their organizations [6]. Only Pareto analyses⁸ appear to be more common.

Such wide reliance on SPC and Pareto analyses is due in large part to the SW-CMM and its heritage in industrial engineering and total quality management, and also because graphical presentations are intuitive⁹. Of course, not all problems have the same solution. SPC, for example, is only one tool, and it is not always used correctly. One size does not necessarily fit all, yet there still is relatively little evidence supporting use of alternative data analytic approaches.

Although the use of designed experiments and quasi-experimentation fits quite naturally into applications of causal analysis and defect prevention [7, 8], they still are not widely used in higher-maturity organizations; fewer than 10 percent reported that they used designed experiments, and only two respondents said they used quasi-experimental designs [6].

There is evidence, however, that experimental methods may be used more commonly than you might think. For example, in a recent study of practitioners and users of software measurement, almost 40 percent said their organizations commonly employ experiments and/or pilot studies prior to the widespread deployment of major additions or changes to development processes and technologies. Undoubtedly, not all of them follow rigorous methodological standards. However, the results are encouraging because the study sample was structured to include representatives from organizations that have had varying success with their software measurement efforts; there were failures as well as successes [9].

One occasionally sees more sophisticated uses of curve fitting, for example, using Rayleigh curve-based models¹⁰. Six Sigma approaches also are gaining increased interest in the process improvement community [10, 11]. However, widespread use remains uncommon [6]. Indeed, there still is minimal use of multivariate methods, classical or otherwise. Even basic statistical methods such as regression analysis or analysis of variance are reportedly used by relatively few high

maturity organizations [6, 9].

Available Guidance

With the exception of SPC, capability maturity model integration models do not offer a great deal of guidance on using data analytic methods. Indeed, even ISO/IEC 15939 [12] focuses more on measurement fundamentals as opposed to detailed guidance about analysis. With the notable exception of the Practical Software and Systems Measurement Support Center's work and their Practical Software Measurement Insight tool¹¹, the same is true for all of the best-known frameworks, models, and standards in the process improvement community. Yet any raw data must be analyzed and interpreted with care.

The early emphasis is on presentation graphics in most software measurement programs, probably because visual displays often appear to be intuitive. But they also are often misused and misinterpreted [13, 14]. The challenge to the measurement community is to balance rigor and methodological defensibility with clarity and practical import. There is often a very real culture clash between measurement experts who are trained to attend to excruciatingly obtuse detail and practitioners who need actionable guidance.

Classic tools for process improvement in the manufacturing world date back at least to Deming [15] and Juran [16]. Techniques such as Pareto charts, run charts, histograms, pie charts, scatter diagrams, bar graphs, and control charts are first principles for any good manager or practitioner in most engineering disciplines. These are not advanced topics by any means, although their proper application and interpretation do take training and experience. Indeed, control charts are often posted on the walls for easy reference in many enterprises, but not often in software organizations, which is something of an anomaly. The software process improvement community often seems to have forgotten its heritage in total quality management and industrial engineering.

Detailed, prescriptive, how-to guidance is outside the province of capability maturity model integration models. But other sources of guidance do, of course, exist. Many other books and articles exist in the published literature on both applied statistics and software measurement¹². And many courses for measurement practitioners also are available.

Summary and Conclusions

A successful process for measurement

and analysis is characterized by decision making that regularly includes data analyses results that are based on objective measurement. Following such a process can help projects and organizations make significant performance improvements in their other software processes and in the products and services that those processes help bring about. Moreover, relying on a well-defined measurement process can demonstrate evidence of business value, thus helping justify continued investment in process improvement and the measurement and analysis activities that support it.

The sophisticated use of measurement and analysis that characterizes high-maturity organizations has been shown to result in substantial added value [17]. Organizations that have attained high-maturity status regularly report notable improvements in measured customer satisfaction as well as better schedule and budget predictability. These businesses also commonly provide evidence of increased staff productivity, as measured, for example, by reductions in development effort per line of code or function point. Also they demonstrate heightened product quality with earlier defect detection profiles and marked decreases in defect density during testing.

Measurement is a key enabler for process improvement and enhanced product quality. An organization with a mature approach to measurement and analysis will have confidence in its abilities to effectively deliver products that meet its customer's needs. Measurement must begin early if it is to reach its full potential, and measurement capability must grow over time. It is difficult for us to conceive of serious software engineering or accomplished management without measurement. The time has come for software and systems development to move toward using the same degree of sophistication in measurement that other engineering disciplines have used for many decades. ♦

Acknowledgements

We owe a great depth of gratitude to many people too numerous to mention here, but special thanks are due to David Card, Jim Curfman, Khaled El-Emam, Wolf Goethert, Will Hayes, Lauren Heinz, David Herron, Seshadri Iyer, Cheryl Jones, Jim McCurley, Jack McGarry, Mark Paulk, Jerome Pesant, Kevin Richins, Janet Russac, Sandy Shrum, Guy Taylor, David White, and Dave Zubrow.

References

1. Gansler, J. S. "Software Evaluations

for ACAT I Programs." The Under Secretary of Defense Memorandum. 26 Oct. 1999.

2. McGarry, J., D. Card, C. Jones, B. Layman, E. Bailey, J. Dean, and F. Hall. Practical Software Measurement: Objective Information for Decision Makers. Boston, MA: Addison-Wesley, 2001.
3. Paulk, M. C., C. A. Weber, B. Curtis, and M. B. Chrissis. The Capability Maturity Model: Guidelines for Improving the Software Process. Reading, MA: Addison-Wesley, 1995.
4. CMMI Product Team. Capability Maturity Model Integration (CMMI®) Version 1.1 Continuous Representation. Pittsburgh, PA: Carnegie Mellon University, Mar. 2002.
5. Zubrow, D. "The Measurement and Analysis Process Area in CMMI." ASQ Software Quality Newsletter 2001.
6. Paulk, M., D. Goldenson, D. White, and M. Zuccher. Unpublished data from a study of high maturity organizations, 2002.
7. Wohlin, C. et al. Experimentation in Software Engineering: An Introduction. Boston, MA: Kluwer Academic Publishers, 2000.
8. Goldenson, D., and R. Stoddard. "The Use of Designed Experiments in Software Engineering." Empirical Studies of Programmers: Sixth Workshop. Eds. W. Gray and D. Boehm. Norwood, NJ: Ablex Publishing Corporation, 1996.
9. Goldenson, D., and K. El-Emam. "What Should You Measure First? Lessons Learned from the Software CMM." Software Engineering Symposium, Washington, D.C., Sept. 2001.
10. Card, D., "Sorting out Six Sigma and the CMM." IEEE Software May/June 2000.
11. Sivi, J. "Six Sigma: An Overview." Pittsburgh, PA: Software Engineering Institute, 1 May 2001 <www.sei.cmu.edu/str/descriptions/sigma6.html>.
12. ISO/IEC 15939:2002. "Software Engineering – Software Measurement Process." 11 July 2002 <www.iso.org>.
13. Tufte, E. The Visual Display of Quantitative Information. Cheshire, CT: Graphics Press, 1983.
14. Goethert, W., D. Goldenson, and W. Hayes. "Scatter, Line, Pie, and Bar: Using Charts to Make a Point." Software Engineering Symposium, Pittsburgh, PA, Sept. 1998.
15. Deming, W. E. Out of the Crisis.

Cambridge, MA: MIT Center for Advanced Engineering Study, 1986.

16. Juran, J. M. Juran on Planning for Quality. New York, NY: MacMillan, 1988.
17. El-Emam, K., and D. Goldenson. "An Empirical Review of Software Process Assessments." Advances in Computers. Ed. M. Zelkowitz. San Diego, CA: Academic Press, 2000.

Notes

1. This text is an abridgment and extension of material that previously appeared in the following: International Function Point Users Group. IT Measurement: Practical Advice from the Experts. Eds. D. Herron, J. Russac, D. Coley, J. Curfman, B. Emmons, and J. Schofield. New York: Addison-Wesley, 17 Apr. 2002: 577-604.
2. We called it the "Measurement Mafia" when we were creating the Measurement and Analysis Process Area for CMMI.
3. See, for example, Emmons, B., and C. Dekkers. "Function Point (FP) Maturity Model: How FP Supports the Capability Maturity Integration

(CMMI) Model." CrossTalk Feb. 2002.

4. The data flow arrows in the figure are modified from the original.
5. This figure is based on a similar one in the CMMI Training Materials, which are not publicly available.
6. The higher capability level generic practices are not included in the staged version of the V1.02 CMMI models.
7. In fact, measurement experts routinely subject their own work to empirical evaluation.
8. For an introduction to Pareto analysis, see Florac, W., and A. Carleton. Measuring the Software Process: Statistical Process Control for Software Process Improvement. Reading, MA: Addison-Wesley, 1999: 62-64
9. Ibid. This is a good source for a useful treatment of SPC in software process improvement.
10. A Rayleigh curve yields a good approximation to the actual labor curves on software projects.
11. The PSM treatment of data analysis focuses on graphical presentations. Their examples range from the very simple to sometimes quite sophisticated uses of multivariate graphical indi-

cators. See McGarry, J., D. Card, C. Jones, B. Layman, E. Bailey, J. Dean, and F. Hall. Practical Software and Systems Measurement: A Foundation for Objective Project Management. V4.0b. Department of Defense and U.S. Army, Oct. 2000, particularly part 4, chapter 3.

12. A few good places to start include Van Solingen, R., and E. Berghout. The Goal/Question/Metric Method. New York: McGraw-Hill, 1999; Fenton, N. E., and S. L. Pfleeger. Software Metrics: A Rigorous and Practical Approach. 2nd ed. Boston, MA: PWS Publishing Company, 1999; and Florac, W., and A. Carleton. Measuring the Software Process: Statistical Process Control for Software Process Improvement. Reading, MA: Addison-Wesley, 1999; as well as the work by Tufté [13], Wohlin [7], and Card [10] cited earlier. Journal articles with examples of data analytic methods worth emulating often can be found in Empirical Software Engineering: An International Journal ["International Symposium on Empirical Software Engineering." IEEE Computer Society Staff, 2002].

About the Authors



Dennis R. Goldenson is a senior member of the technical staff in the Software Engineering Measurement and Analysis group at the Software Engineering Institute. His work focuses on using measurement and analysis in software engineering, the improvement of process appraisal methods and models, and the impact and transition of software process improvement and other software engineering practices. He is a principal author of the "Measurement and Analysis Process Area for CMMI" and served as co-lead of test and evaluation for the project.

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890
Phone: (412) 268-8506
Fax: (412) 268-5758
E-mail: dg@sei.cmu.edu



Joe Jarzombek is director of Software Intensive Systems in the Office of the Under Secretary of Defense (Acquisition, Technology and Logistics). He serves there on loan from the Institute for Defense Analyses in Alexandria, Va. A project management professional, certified by the Project Management Institute, he previously served as vice president of Product and Process Engineering, for the USERTRUST Network, a Public Key Infrastructure that provides security and legal protections for Internet transactions. Prior to his retirement from the military, Jarzombek was a lieutenant colonel in the U.S. Air Force, he was manager of the Air Force's Computer Resources Support Improvement Program Office, and he was sponsor of CrossTalk, The Journal of Defense Software Engineering. He also is a principal author of the "Measurement and Analysis Process Area for CMMI."



Terry Rout is a senior lecturer in the School of Computing and Information Technology at Griffith University, Queensland, Australia, and is associated with the Software Quality Institute there. He lectures in the areas of software engineering, software quality, and project management. Rout is chairman of the Australian Committee for Software Engineering Standards, and has been a member of the Australian delegation to the International Committee on Software Engineering Standards since 1992. He is the project editor for ISO/IEC TR 15504-Software Process Assessment. Rout also is a member of the newly established Capability Maturity Model Integration (CMMI®) Expert Group, which advises the Software Engineering Institute on issues related to the CMMI adoption and interpretive guidance.