CrossTalk

The Journal of Defense Software Engineering

Sponsored by the
Embedded Computer
Resources Support
Improvement Program
(ESIP)

June 1999
Volume 12 Number 6

Published by the
Software Technology Support Center

The Need for a Measurement and Analysis Process: Focusing on Guidance for Process Improvement

Lt. Col. Joe Jarzombek
ESIP Director

As software and systems engineering disciplines continue to evolve and become more integrated, *measurement and analysis* as a support function becomes a basic practice. It is required by management activities, such as project planning, monitoring, and control. As an organization matures, objective management becomes a common practice. Basic project management indicators—cost, milestone completion, defects, etc.—are augmented by process management indicators, such as process change impact and process performance. Measurement and analysis (M&A) supports these management activities. M&A has been important for organizations striving for higher levels of maturity and continuous improvement of processes, products, and projects. Measurement has been recognized as a key enabler for performance-based management. Indeed, to become more competitive and to strengthen their ability to more quickly achieve higher maturity, some organizations have created an additional process area for measurement when using existing models to guide their process improvement efforts at lower levels.

Measurement helps organizations and decision makers by providing meaningful information regarding the quality, adequacy, and evolutionary progress of processes, products, and projects. Measurement offers the insight needed to plan, control, manage, and improve:

- the product technical adequacy and performance.
- its schedule and progress.
- resources and cost.
- growth and stability.
- product quality.
- lifecycle process performance.
- technical effectiveness.
- customer satisfaction.

In today's Department of Defense (DoD) "acquisition reform and outsourcing" environment, defense organizations and project offices are encountering more complex risk management responsibilities, diminishing organic resources, and more reliance on commercial products and processes. Information technology legal requirements demand results-based mission improvement and process improvement. Integrated program management is needed, and it is best supported by a measurement program shared by the acquiring and delivering organizations.

How measurement and analysis is represented in any Capability Maturity Model® (CMM®) that guides process improvement is of vital concern. The Federal Aviation Administration (FAA), working with the Software Engineering Institute to integrate software, systems engineering, and acquisition disciplines into a single model (dubbed iCMM), has specified M&A as a separate process area. The Office of the Secretary of Defense-sponsored effort to integrate software, systems engineering, and integrated process and product development (IPPD) into a single model (dubbed CMMI) has also adopted M&A as a distinct process area in the draft released for stakeholder review. This is significant because historically, the measurement process was not explicitly defined in single-discipline models. Many assessors have indicated that M&A is a common problem among assessment findings for those organizations that do not have a measurement program in place.

To provide appropriate guidance to incorporate M&A into any model that supports process improvement efforts, four enablers need to be considered:

- Provide high visibility of the M&A process. In the absence of overt guidance, M&A activities are independently created. This lack of coherence significantly impedes an organization's move to higher maturity practices. As organizational processes are developed, earlier M&A processes must be rewritten if no coherent guidance was provided in earlier phases of organizational evolution.
- Provide a simple process with a sequential set of specific practices that focus upon providing indicators that satisfy information needs, which have been derived from business goals and objectives.
- Provide guidance for the growth of the M&A program. As an organization matures, the nature of the M&A process evolves. The nature of the goals changes from simple visibility into what is happening to visibility into the impact of process changes. Analysis methods change from simple fish-bone charts to detailed root-cause analysis. The nature of the data available from the collection process changes from major milestone visibility into detailed subprocess performance. The collection process may change from manual collection and simple spreadsheets to more complex automated data collection tools.
- Clarify the relationships—cause-effect, output-input, terms-definitions, etc.— among the various process areas. Practices in the M&A process support other processes in that they require M&A to be effective. As a separate process area in a CMM, M&A creates a tremendous opportunity for clarification, and it supports conformance with ISO 15504, which requires assessment of M&A.

Regardless of what model might be used, the explicit incorporation of M&A as a distinct process area should provide the management visibility and focus that organizations have needed to guide their process improvement efforts. Use of M&A, as a separate process area with practices emphasized early in project, product, and process evolution, should enable organizations to more quickly achieve quantitatively managed processes and better products.◆

Beware the Unacknowledged Source

I recently spoke with Bob Grady, who showed me a letter he had written to you regarding my article "Metrics Problem Solved?" (CROSSTALK, December 1997). He pointed out the similarity of the "Codex Metrics" in my article to his Figure 10-3, "Software Metrics Etiquette," *Practical Software Metrics for Project Management and Process Improvement* (Prentice-Hall, Upper Saddle River, N.J., 1992). I acknowledge that his work is undoubtedly the original source of this information and to say that I was extremely embarrassed and shocked is an understatement. I had no idea I had plagiarized his work. By necessity, we build on the work of those who come before. The credit to an author is the insight that they bring to previously published work or words, not in stealing from other authors. The problem is that I am exposed to so much information that after a while I am not sure how or where a concept originated. But because of the obvious similarity between my words and the original, it appears that I am the perpetrator of metaplagiarism.

My Victorian forebears would call this a cautionary tale, worth repeating for CROSSTALK readers and contributors alike. The message is, "beware the unacknowledged source." I concur with Mr. Grady's words (which I paraphrase slightly): An unacknowledged reference, much less a restatement of the essence of any work without proper framing of how such a restatement adds to the original contribution, belittles the original.

I apologize to Mr. Grady for my infraction.

David R. Pitts
Phoenix, Ariz.

Coming Events

13th Annual Ada Software Engineering Education Team (ASEET) Symposium

Theme: Ada in the 21st Century: Academic, Government, and Industry Perspectives
Dates: July 26-29, 1999
Location: Colorado Springs, Colo.
Registration and conference information is available at http://www.acm.org/sigada/aseet/

14th Annual ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA '99)

Dates: Nov. 1-5, 1999
Location: Denver, Colo.
Topic: OOPSLA offers a collage of technical papers, practitioner reports, topical panels, outstanding invited speakers, exhibits, poster, demonstrations, formal and informal educational symposiums, as well as an exceptional tutorial program, and plenty of social opportunities to mingle.
Contact: Brent Hailpern, conference chairman
Voice: 503-252-5709
Fax: 503-261-0964
E-mail: oopsla99@acm.org
Internet: www.acm.org/sigplan/oopsla

Software Testing Analysis & Review STAR '99 West

Theme: Improving Software Testing and Quality Engineering Practices Worldwide
Dates: Nov. 1-5, 1999
Location: San Jose, Calif.
Sponsor: Software Quality Engineering
Topics: Specific ways to improve testing efforts and results. Field-proven techniques for testing client-server, object-oriented, GUI, and Internet applications. How to use test engineering to consistently achieve greater software quality. The best Internet/Web testing tools and how to use them effectively. How to lower development costs and boost productivity with test engineering.
Voice: 1-800-423-8378 or 904-278-0707
Fax: 904-278-4380
E-mail: sqeinfo@sqe.com

The Sixth International Symposium on Software Metrics

Dates: Nov. 5-6, 1999
Location: Boca Raton, Fla.
Theme: "Taking the Measure of New Technology"
Topic: The application of measurement (through empirical studies) to understand and manage new software technologies, including their related tools and processes, such as commercial-off-the-shelf-based development and Web-based applications.
Contact: David Card, general chairman, Software Productivity Consortium
Voice: 703-742-7199
Fax: 703-742-7200

On the cover: A fighter pilot preparing a jet for flight, Hill Air Force Base, Utah, circa 1960 illustrates this month's theme of measures and metrics. Turn to pages 4, 8, and 12 for related stories. Photograph courtesy of Dave Kendziora, Ogden Air Logistics Center historian.

An Effective Metrics Process Model

Capt. Thomas Augustine, *U.S. Air Force*
Charles Schroeder, *Colorado Technology University*

Here, we describe how a number of Air Force Space Command bases determine the effectiveness of metrics within their organizations. Participating in these studies were communicators from Falcon Air Force Base (AFB) (now Shreiver AFB), Colo., Peterson AFB, Colo., and Malmstrom AFB, Mont. Though limited to communications and information metrics, this process could be applied to any organization that requires decisions to be made based on facts rather than made haphazardly.

MANY ORGANIZATIONS TAKE measurements or metrics because they have the capability to measure, rather than determining why they need the information. Unfortunately, measurement for the sake of a number or statistic rarely makes a process better, faster, or cheaper. A poor measurement can hurt a process if incorrect decisions are based on the result of that measurement. People at all levels of organizations continue to take measurements hoping that they will shed light on the best way to provide a product or service. Though fraught with good intentions, these poorly contrived measurements add to the confusion of what should and should not be measured.

Metrics Process Model

Until a year ago, many of the communications and information metrics of Air Force Space Command (AFSC) were taken because they had been collected for years, and people thought those metrics must have a purpose.

At that time, many metrics were not being used to make a decision based on fact, but fulfilled a headquarters' requirement to report on information by a certain date every month. After a fairly extensive study, the AFSC Senior Communicator (SC) changed the format and collection of many of these metrics, while deleting the requirement for many that had little value.

Like many discoveries, the process for metrics collection and analysis in this directorate was the result of a change in leadership. Communications metrics at AFSC seemed to provide good information, since senior leaders did not complain about content or format of the 30 metrics collected at the headquarters level. Haphazard metrics collection continued until a number of new senior leaders asked why these metrics were being collected and if they were the right measurements for their organizations. These questions sparked a complete review of the metrics collection, analysis, and reporting process.

After completing a thorough analysis of existing approaches and an analysis of literature on this topic, we decided on a common definition and set of criteria necessary in good metrics collection, reporting, and analysis. The process derived from this research is noted in Figure 1.

Foremost in our quest for good metrics was a definition of a "good metric." Although a review of current literature on metrics indicated many definitions of this term, they could be summarized as one that helps the right level of leadership make the right decisions in a timely manner, based on fact rather than "gut feeling."

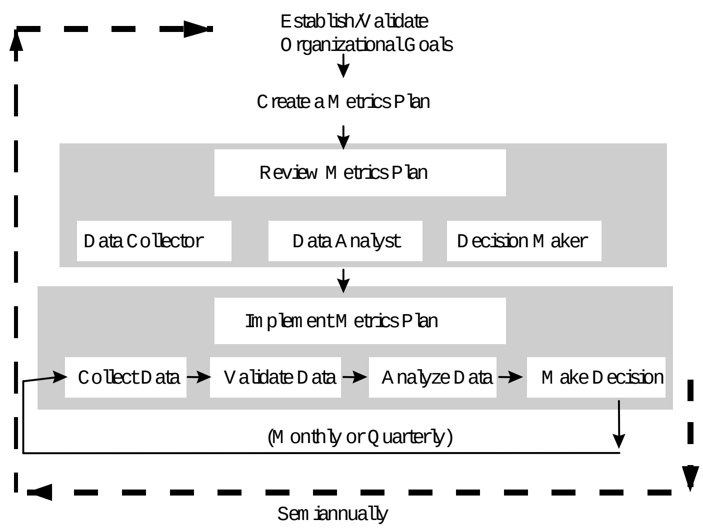


Figure 1. *Metrics process model.*

Applying the Model

Establish and Validate Organizational Goals

With that definition in mind, the majority of authors studied noted that the first step in good metrics collection is understanding the goal. Rather than ask what should be measured, ask what is important to the organization and its customers.

Many organizations have trouble with this; however, the Communications and Information Directorate at AFSC did a thorough review of its customers' requirements and understood what was important to the organization's success. The SC directorate validated its organizational goals and objectives with its customers, suppliers, and senior managers, when it published its strategic plan. Re-validated semiannually, this eight-page document outlines the direction the unit is expected to take in the next few years. Notably missing from the organization's strategic plan was a link of metrics to measure the progress of these goals.

In re-validating this strategic plan, using metrics as a tool to measure these goals, many people noted that the goals were too general because they could not be measured. These goals and objectives were revaluated, ensuring that each objective had an associated measurement to ensure progress.

Management Issues

Although these goals are important to every organization, it can be difficult to focus on defining clear, measurable goals, based on what is important to customers. Senior management can be skeptical about the value of spending time defining such goals. The Communications and Information Directorate at AFSC understood the need for such goals but proceeded cautiously, defining those goals that were most easily quantified first.

Measures of a system's up-time rates and availability were clear targets with measurable rates and long data histories. Once these goals were proven to provide useful decision points, senior leaders were willing to define other goals of interest to the organization and ultimately to the customer. Each organization must decide how many goals it needs to effectively manage its resources and meet its customers' requirements. Through trial and error, the organization found that its customer requirements could be encapsulated into about 10 measurable goals and 40 more specific subgoals called objectives. The goals provided a broad-based definition for what was important to the organization, while the objectives specified actions necessary to meet customer requirements. Each objective was written so as to be clearly measurable, and at least one associated metric was created for each objective to provide decision-making information to senior management.

Every organization will have a different approach to establishing goals based on customer requirements, but regardless of the approach, it is important that these goals are measured and quantified in terms that senior management can understand and fully support.

Create a Metrics Plan

The Communications and Information Directorate had a strong data collection program, but the analysis and use of this information was limited. Although the intent of these metrics was to measure an important or problem area, the number of metrics continued to grow, while the analysis was almost nonexistent.

A plan was created to validate the purpose of each metric. Rather than modify existing metrics, the metrics program needed an overhaul. Many of the cost, schedule, and performance metrics were relevant because they directly measured the mission. However, the metrics process to collect and analyze this information required updating. We defined an overall metrics philosophy as an adjunct to the strategic plan and noted that each new metric had to have basic information associated with it, making it useful to the right people at the right time. Figure 2 is a form we used to collect this information in a single, neat package so everyone from collectors to decision makers could understand their purpose in collecting, reporting, and making decisions based on this metric. Although simple, this broad overview causes people in the organization to think before creating or approving a metric. It also marks the conditions under which the metric will remain useful. This makes the process easier for semiannual review of the metrics, because the criteria are spelled out and metrics that have outlived their usefulness are deleted or replaced.

Metric Title	Brief Description	
Link to Goals/Objectives	Decision(s) based on analysis	Who makes decision(s)
Who collects data	How is data collected	How often is data collected
Who reports data	How and to whom is data reported	How often is data reported
Who analyzes data	How is data to be analyzed (formulas and factors)	
Lowest acceptable values	Highest acceptable numerical values	Expected values
At what point will you stop collecting this metric		

Figure 2. *Metrics collection form.*

Review Metrics Plan

In creating this metric plan, we noted that there may be other factors that we had not considered when defining each metric. To review the data objectively, we surveyed our data collectors and senior leaders to see if they understood why we collected each metric. The results were enlightening and helped to create a usable metrics program. In this survey, we asked questions about the metrics' perceived usefulness, its ability to aid in decision making, goal of the metric, tolerances set for the highest and lowest acceptable values, and timeliness of feedback based on analysis of the data. We could have interviewed people instead of taking a survey but believed anonymous answers would be more honest. We distributed one survey for each metric to each of three groups. One survey was given to data collectors and another to senior managers assigned to make decisions on each metric. The third set of surveys was distributed to the metric owners who designed the metric and are assigned to analyze the data. These surveys were used as the control group because those who designed the metric should understand why the metric is important in making decisions.

Through this survey, we obtained raw data on specific problems and accolades associated with each metric. Although we addressed specific problems, our primary reason for the analysis was to assess the overall usefulness of our metrics program. This analysis, though useful to every level, was of greater use to senior managers who make final decisions on the types of metrics to be collected in the organization.

The first trend analysis showed that one-third of the metrics were not useful in making decisions. Some had no reason for being collected; others had outlived their usefulness. Also noteworthy—few people received timely feedback from those who were assigned to analyze the data. All of these factors led to a metrics program that provided a lot of data but little useful information. Before implementing this metrics plan, many believed that their metrics were the "right" measurement.

Changes were made to existing metrics to streamline and standardize collection processes, and a number of metrics were deleted. After the new metrics passed a trial period, senior managers were confident that the new metrics provided information necessary in making decisions.

Implementing a Metrics Plan

A plan is proven only when it is implemented. Senior managers realized this, and after careful planning, proceeded to provide policy and process clarification to those collecting and analyzing data.

Policy and Process Issues

Gathering and quantifying information initially takes considerable effort but eventually becomes a regular facet of the organization. Although a metrics plan can detail how to collect data, only people can collect and analyze the right data. In gathering metric information, AFSC had to overcome many logistical concerns not only in getting the data but also in ensuring that the data was consistent among the nine communications agencies for which this organization compiled information. They began by clearly defining the requirements in a policy letter signed by the SC. Information to be collected and suspend dates for collection were defined in this policy, which each of the nine communications organizations were required to follow.

Once this policy was signed, the task of ensuring consistent, measurable data had just begun. Though the organization felt that its policy and direction was clear, it took three months for all data collection agents to consistently collect the information requested. After a series of clarifications and minor changes in the collection process, a consistent process to collect metric data was defined and published. Although different for each organization, it can be assumed that even with the best intentions, consistent data collection is an iterative process requiring modifications until all data collectors use the same processes and methods. Although automation can help in this consistency, it is ultimately up to the people who define the metrics to clearly articulate the process for data collection.

Metric Utilization by Management

Even if a metrics plan were perfectly implemented, it still would be incomplete unless the correct level of management makes decisions based on the metrics. It has been well-documented that management buy-in and commitment are necessary before a metrics process can work.

AFSC ensured that its senior management understood the implications of the metric analysis through monthly metric meetings with senior managers, midlevel managers, and people who collect and analyze the data. This type of high visibility is crucial for a successful metrics program. Without definite due-dates and justification for information collection and analysis, senior managers likely would not make metrics a priority. Everyone who collects, analyzes, or makes decisions based on metrics data must be aware of the process, due-dates, and most important, that the metrics are being used to make corporate decisions. When all parties involved understand the importance of these metrics, they are likely to make an effort to collect accurate data, analyze it, and ensure reporting is done quickly to aid in the decision-making process.

Reviews

To be effective, even the most perfect plan needs consistent review. The first review of the metrics plan for this organization shook up the way we used metrics to make decisions.

After the initial review, there was a large turnover in senior leaders, changing some of the primary goals and focuses of the organization. There was another review at the semiannual point, and although the changes were much more subtle, metrics were again changed to reflect the criteria needed for solid decision making within the organization. This continues to be an iterative process, and the senior leadership of the AFSC SC's office is committed to continuing this process.

From Model to Reality

Although there were a number of positive examples using this metric plan, the metrics depicting network status had particularly good results. The SC was measuring up-time rates on servers, and although senior management realized that these servers were a key to our success and mission accomplishment, they did not have well-defined goals.

In starting this project, no one knew exactly why such metrics as "up-time rate of servers," "numbers of computers in an organization," and "number of megabytes of data processed" were collected. These measurements were discarded because they were only one-dimensional, leaving the data analyzer and decision maker with such questions as "Is that a good up-time rate?" and "Is that a lot of data being processed?"

The right measurements soon became apparent in the goal-definition stage. Originally, goals were stated solely in terms of up-time rate and easily measured quantities—not because these were the best metrics, but because they were the easiest to collect. Many metrics originally were turned down because they were not easily placed in a bar or Gantt chart. It soon became apparent that by defining the goal, the metric becomes obvious, rather than defining an easy metric and trying to make a goal based on it.

After much deliberation, the goal became "reduce operations and maintenance costs by 20 percent while maintaining equal or better service to the customer."

With this clear, measurable goal in mind, metrics were created that measured total system cost, cost per capita, and cost per megabyte of data. Cost was defined in terms of dollars and manpower required. The purpose of this goal was clear, and the decisions associated with these measurements were no longer nebulous. These costs could be compared to in-house and contract labor costs. This organization found that the most useful metrics were those that compared two or more quantities rather than solely reporting finite measurements. When these metrics were compared with the up-time rates, some excellent savings opportunities were discovered.

By asking why the metric is important to you, is everyone consistently measuring the same type of data, and how will decisions be made based on the data, collection data became clear, concise, and consistently repeatable. Decisions could effectively be made from the compiled information.

A number of important decisions were made based on the new metrics. For example, in looking at network status and up-time rates on servers, it was determined that a 100 percent up-time rate was not cost effective, based on the analysis of the cost of up-time vs. network availability and efficiency.

Also by comparing costs per capita with costs per megabyte, many decisions were made to consolidate information processing operations, again saving maintenance man-hours and server costs. By following this systematic process, the organization was able to define clear, measurable goals and obtain information crucial to the decision-making process.

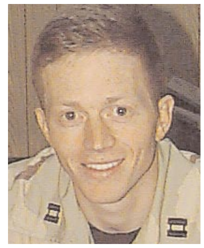
Conclusions

Many people may look at this model or method and note its simplicity. Throughout the literature analyzed, however, authors note the difficulty of creating metrics that are easy to understand yet help the right level of management make timely decisions based on fact. Many times, the difficulty is that we continually ask how to measure a process rather than determining what decisions need to be made.

If organizational goals are written clearly and are measurable, creating a metrics program becomes simple. A successful metrics program ensures that data is collected and analyzed consistently, and most important, this program ensures that the right people are making timely decisions based on fact. All that remains is a semiannual review to ensure that you stay on track with the decisions your organization is making based on these metrics.

We encourage you to take this analytical view of your metrics, thinking not of individual measurements but of a system that helps you make good organizational or corporate decisions. This process has been proven throughout the available literature and in practice. Most organizations could benefit from implementing a structured metrics program. ◆

About the Authors



Capt. Thomas Augustine is the Mission Systems flight commander in the 341st Communications Squadron at Malmstrom Air Force Base, Mont. He has held many leadership positions in the Air Force and has a master's degree in administration from Central Michigan University and a master's degree in computer science from Colorado Technical University.

Capt.Thomas Augustine
Building 500
Malmstrom AFB, MT 59402
Voice: 406-731-6785

Charles Schroeder is professor of computer science at Colorado Technical University in Colorado Springs. He has extensive university-level teaching and administrative experience and consulting and contracting experience with government organizations. He has a bachelor's and a master's degree from Southwest Texas State University and holds a doctorate from Texas A&M.

Dr. Charles Schroeder
4435 N. Chestnut Street
Colorado Springs, CO 80907
Voice: 719-590-6761
Fax: 719-598-3740
E-mail: cschroeder@cos.colotechu.edu

Measures and Metrics Web Sites

www.sei.cmu.edu/sema/welcome.html

This is the home of the Software Engineering Measurement and Analysis team of the Software Engineering Institute (SEI). The focus is on software measurement and empirical research that accelerates the identification and adoption of improvement to software engineering practices.
The site contains valuable information about SEI's technical reports, guidebooks, and training regarding software measurement.

www.psmsc.com

The PSMSC site is the home of the Practical Software Measurement Support Center. This site includes information about the PSMC. A section is devoted to current issues and news. It also includes a complete copy of *A Guide to Objective Program Insight*, one of PSMC's valuable products.

www.stsc.hill.af.mil/Metrics/index.html

The Software Technology Support Center's metrics page is dedicated to help organizations improve their metrics programs as it relates to software process improvement. Topics such as Practical Software Measure and Evaluating Measurement Capablilty are available. A great list of recommended readings also can be found.

www.ifpug.org/home/docs/otherpages.html

This is the home of the International Function Point Users' Group. It contains valuable links to useful function point-related tools as well as more than two dozen links to metrics-related sites.

PSM Insight: The Army-DoD Tool to Implement Issue-Driven Software Measurement

Don Scott Lucero
U.S. Army Software Metrics Office

The Army Software Metrics Office has developed a software measurement tool, PSM Insight, to implement the Army's issue-driven software measurement policy [1] and the practical software measurement (PSM) process [2]. The PSM Insight tool supports tailoring and controlling the data needed to implement an effective measurement program. This article describes some of the challenges faced by the PSM Insight Development Team to deliver a tool with the flexibility to support a tailored measurement process.

The PSM Process

PSM is a Department of Defense (DoD)-sponsored project to provide program managers with the objective information needed to successfully manage software-intensive projects. PSM is based on software measurement experience with DoD and industry projects. Measurement professionals from DoD, industry, and academia have collaborated to define best measurement practices used within the software acquisition and engineering communities.

PSM treats measurement as a flexible process, not a predefined list of graphs or reports. The process is adapted to address the specific software issues, objectives, and information requirements unique to each project. The PSM process is defined by a set of nine best practices, called measurement principles. The underlying objective of the PSM guidelines is to integrate the measurement requirements into the software process. Software measures are tailored to reflect the existing project management and software development processes, ensuring that the measures provide meaningful and cost-effective results. The measurement process also is integrated with existing risk and financial management processes to provide a basis for objective decision making.

The PSM process is defined in the PSM Guide, *Practical Software Measurement: A Foundation for Objective Project Management*. The guide explains the basic concepts of the software measurement process, offers detailed implementation guidance, and provides realistic case studies of software measurement used on typical projects. In addition to the guide, the PSM project provides training and workshops on the PSM process, hands-on management support as requested by other projects, and the PSM Insight tool to manage data.

Development of PSM Insight started in February 1997 with a Memorandum of Agreement between the Army Software Metrics Office and the PSM Steering Group. This agreement was to develop a single tool to support an integrated DoD software measurement strategy. PSM Insight provides a PC-based management capability to implement the PSM process. PSM Insight guides managers through tailoring software measures to their specific project and provides the data management capabilities to select and monitor project-specific indicators, measures, and data items.

The PSM Guide presents a systematic but flexible, issue-driven measurement process with examples of data items, issues, categories, and measures. PSM Insight supports the PSM process by importing, storing, and graphing these minimum examples, but also provides a flexible database that allows a user to define project-specific data items, issues, categories, and measures. The PSM Insight tool provides desktop support for a software measurement project that achieves the nine fundamental principles of the PSM process, as illustrated in Table 1.

The PSM Insight Tool

PSM Insight is a Windows-based application that allows a high level of flexibility in data management including data modification, data browsing, and sophisticated graphing capabilities. PSM Insight has the capability to tailor software measures to unique project issues and allows using data already available from an existing software development process. The tool has been designed in compliance with industry standards for Open Database Connectivity, which allows the tool to create or automatically access different databases with dissimilar data formats.

PSM Insight can accept any unique data parameter required for a software measurement project and can manage data according to the attributes and software components,

Table 1. *The nine fundamental principles of the PSM process.*

- Project issues and objectives drive the measurement requirements.
- The developer's software process defines how the software is measured.
- Collect and analyze data at a level of detail sufficient to identify and isolate software problems.
- Implement an independent analysis capability.
- Use a systematic analysis process to trace the measures to the decisions.
- Interpret the measurement results in the context of other project information.
- Integrate software measurement into the project management process throughout the lifecycle.
- Use the measurement process as a basis for objective communications.
- Focus initially on project-level analysis.

organizations, or activities.

Data Items, Attributes, and Structures

A data item is a specific type of information collected to manage and monitor a project. Data items quantify what is being measured. Common data items include "Start Date," "End Date," "Number of Test Cases," "Cost," "Number of Requirements," and "Defects."

Attributes are characteristics or properties of a measure that distinguish one data item from another. Common attributes include "Planned/Actual," "Version," or "Language."

The structure defines the organization of a project and identifies the level at which data items are collected. For example, this could be a "Software Process Activity" structure of work task that must be completed in the project, a "Software Component" structure of software products that make up the entire project, or "Lines of Code" collected at the organizational level or by functional unit.

Because of the size limitations of this article, only a few examples of the PSM Insight tool displays are given in Figures 1 through 3. Figure 1 shows the Main Menu Window of PSM Insight, which allows the user to create project-specific software measures. This display provides the user with three icon groups to define the tool's top-level functions: Select the Project, Tailor the Measures, and Apply the Measures.

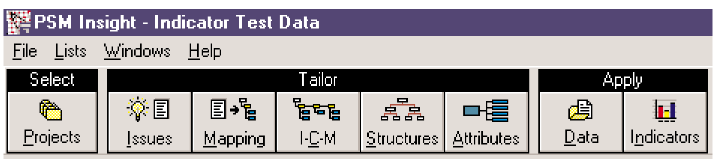


Figure 1. *Main Menu Window of PSM Insight.*

The Projects icon, in the Main Menu Window, allows the PSM Insight user to define or select a project on which software measures are collected.

The Tailor icon group allows the user to define and tailor the software measurement project. It includes five functional icons: Issues, Mapping, I-C-M (Issues-Categories-Measures), Structures, and Attributes.

The Project-Specific Issue List, shown in Figure 2, allows the user to list all issues that are critical to the success of the project.

The Indicators function allows the user to view graphic displays of the measurement data, shown in Figure 3.

For a full description of PSM Insight tool displays, visit the PSM Web site at www.psmsc.com.

The Challenges of Building PSM Insight

The Open Requirements-Definition Process

PSM Insight is being developed in a rapid-prototyping process using Borland's Delphi, Version 2.0. This environment allows a nonproprietary, run-time program to be created for Windows 3.1, Windows 95, and Windows NT. The Windows-based interface to the advanced Delphi design capabilities provides a powerful tool for information analysis and retrieval.

The initial requirements for PSM Insight were defined from the Army's experience in applying previous metrics man-

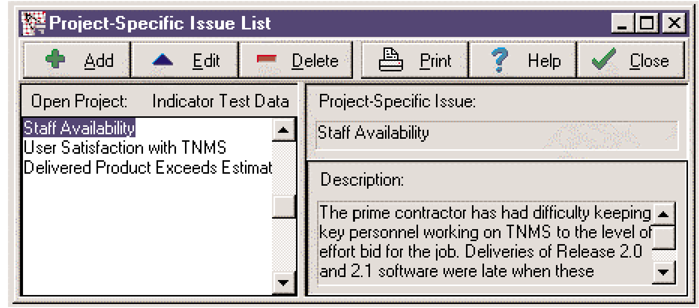
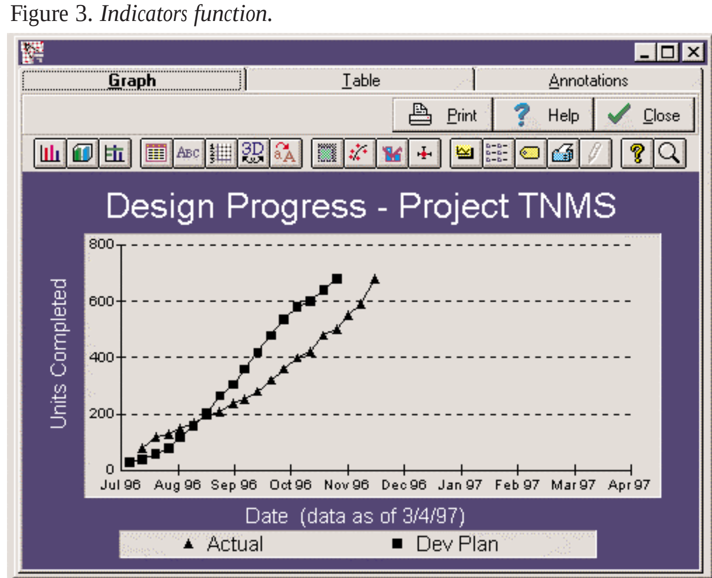


Figure 2. *Project-Specific Issue List.*

agement tools, including Metrics Guided Maturity (MGM) and Software Metrics Management Information System (SMMIS). PSM Insight requirements initially were defined through internal meetings of the PSM Insight Development Team. The process to define user requirements greatly expanded after the first year of development, when the first prototype was available for demonstration. PSM Insight became the subject of a requirements-definition workshop at the first PSM User's Group Conference. The workshop allowed participants to evaluate the existing PSM Insight requirements and recommend new attributes for future versions. The challenge the development team faced was how to reach a consensus on the requirements baseline, while allowing any interested party to suggest new features or enhancements to the tool. However, the contributions from these open workshops have proven invaluable for focusing attention on common areas of concern and for prioritizing future design work.

The challenge in the open requirements-definition process is how to effectively and diplomatically handle requests that conflict or interfere with each other. For example, a common conflict in user expectations is the level and complexity of the security features of PSM Insight. User requests range from completely open access to all PSM Insight features and data, to restriction of access based on multilevel roles in an organization. This open requirements-definition process has also created more work for the PSM Insight Development Team. The team had to tabulate, evaluate, and assign priorities for each new tool

Figure 3. *Indicators function.*



requirement. The team implemented the highest-priority requirements first, while incrementally building essential capabilities required in the objective system.

Supporting Changes to the PSM Guide

PSM Insight's formal requirements base, the PSM Guide, is reissued about once a year. Terminology, definitions, and emphasis in the guide are modified to address changing needs and lessons learned in the measurement process. An important feature of PSM Insight is to provide the approved PSM Guide's tree of successive I-C-Ms as templates during the tailoring process. A user may select, rename, edit, remove, or add to the I-C-M templates that are suggested by PSM Insight at each step. These items are embedded in PSM Insight as text descriptions, preset value lists, and online help. The recommended I-C-M templates are further described through training examples and sample data sets in the tool. Because of the extensive use of embedded descriptions, the PSM Insight Development Team is faced with a major challenge each time the I-C-M templates are changed. During the development effort, changes in the PSM Guide have caused entire issues, categories, measures, and measurement elements to be renamed, removed, or redefined.

Another PSM Insight design challenge has been to allow the tool to be tailored to each user, such as providing the capability to create and modify project-specific I-C-M templates. By allowing the tool to be tailored, the challenge has been to provide the ability to define and refine the way an evolving project is organized. The development team is trying to determine how to build measures that span multiple projects. Tailoring a measurement program tool to organizational structures requires the capability to combine information across multiple projects that may be configured differently, with data collected at different levels and reported at various frequencies.

Meeting the Users' Technical Capabilities

A major development challenge of PSM Insight has been to build a tool that will be useful for both the novice user of measurement data and the experienced manager. The PSM Guide directs a user through a process of tailoring software measures to specific project issues. This process starts with a defined set of six "common issues" that apply to any project. The PSM Guide then assigns each common issue a set of "issue categories" that are linked to appropriate measures. A primary requirement of the tool is to lead users through the same scheme of I-C-Ms.

However, this process is a nuisance to experienced managers who already know the measures and data they need to collect. The PSM Insight graphical user interface requires the user to walk through the I-C-Ms scheme that is prescribed in the PSM Guide. This may result in an additional step for the experienced manager and therefore violates a PSM Insight design objective that additional effort in managing data must be minimized. A related design objective is to minimize entering and storing repetitive, placeholder data.

The question remains whether the conceptual model is adequate to reflect real-life measuring and reporting needs. PSM Insight uses structures, attributes, and data items to define a measure. Thus far, these data types have been sufficient to define the measures in the guide and user-defined measures. As the PSM Guide moves from project-level measures to include enterprise- or organization-level measures, the PSM Insight Development Team will need to assess if additional constructs are needed.

Creating Flexibility

PSM Insight needed to retain simplicity and ease of use, while offering the flexibility required for a tailored measurement process. More flexibility frequently requires more options to be made available to the tool users, rather than fixed by the tool designers. However, the number of decisions may become overwhelming. In response, PSM Insight makes extensive use of default values to minimize the administrative burden of answering every possible tailoring option.

Another design flexibility issue is allowing a user to select a specific display view for the measurement data. For example, any data can be presented as either an interval measurement (number of problems opened each month) or a cumulative number (all problems opened to date). To assist in a tailored measurement process, PSM Insight can translate between the two representations when a user selects a specific graph. Measurements that are stored in PSM Insight as "quantity-per-time-interval" can be viewed as "total-to-date" when needed. Using stored data in arithmetic calculations is a complex design problem, especially when aggregating lower-level data upward.

Training Users

Every software application must achieve a balance between advanced, complex capabilities and the ability of a user to understand and work the program. The data-handling flexibility and capabilities for local user customization have increased the complexity of PSM Insight. Subsequently, the PSM Insight Development Team had to design resident features to support usability of the tool. Two methods were implemented: extensive help screens and portable training.

PSM Insight can be a challenging package to learn. Most users take a one- or three-day class on the PSM method to acquire a working knowledge of the terminology and principles behind the software measurement process. On-site workshops are the best way to learn PSM Insight. Expert instructors teach the basics of PSM Insight, provide help in applying PSM to local projects and tailor the PSM Insight software to site-specific needs. For large sites, this is the optimum learning experience. For smaller sites and one-person operations, the PSM Insight Development Team is building a computer-based tutorial (CBT). The multimedia tutorial will review terminology, illustrate the basic features of PSM Insight, present a demonstration of each major function, and provide guided practice sessions for interactive learning opportunities. The CBT simulates PSM Insight as it walks the student through the major features of the tool, providing tips along the way and prompting the student for input during practice sessions. Although a tutorial cannot address site-specific questions, it can help the self-motivated user become familiar enough with the software to explore more advanced features independently.

Using Commercial-Off-the-Shelf (COTS) Products

The Rapid Application Development environment of Delphi allows the PSM Insight implementers to use tools that are more intuitive and visual. In addition, the tools and components can be extended to include third-party components, "widgets," and COTS products.

PSM Insight development has increased code reusability and programming productivity with the use of third-party components and COTS products. Many of PSM Insight's important features have been implemented using these products. However, COTS products introduce their own set of problems and challenges. Listed below are some of the COTS products and third-party components now in the tool, followed by the technical challenges and problems faced when implementing the components.

COTS products and third-party components

- Graphing of Indicators is implemented using Graphics Server from Pinnacle Publishing. Graphics Server is a comprehensive graphing toolkit consisting of a core set of graphing routines that PSM Insight accesses through library functions. Several hundred graphing parameters can be passed to Graphics Server to control displaying, storing, and printing of graphs.
- The Tailoring Reports capability has been implemented with the help of Seagate Crystal Reports. Crystal Reports is a powerful Windows reporting tool that helps PSM Insight dynamically tailor reports to the user's needs. The PSM Insight Delphi code accesses Crystal Reports' dynamic link library (DLL) for sophisticated report generation and printing capabilities.
- The outline lists used in the tool were implemented with a native Delphi component from a third-party developer. This compiled unit gave programmers access to many properties and events to control the display and action of the outline lists.
- The search data and sort data tasks were implemented with third-party components, saving many hours of programming time.

Technical Problems — distributing and installing the run-time DLLs associated with the COTS products:

- Different installations routines had to be provided for 16- and 32-bit environments.
- Identifying when to overwrite existing DLLs had to be defined, since other applications may depend on them.

Technical Problems — saving, storing, and recalling parameters associated with each COTS product:

- PSM Insight must be able to save, retain, and pass data to the COTS products for a seamless integration of the products for the end user.
- Since each DLL is a special type of executable file or application, it has its own set of bugs that have to be handled or avoided. Therefore, code was written to tell PSM Insight what to do when a call to a function fails.
- Vendor support for bug fixes and product upgrades caused some schedule delays.

Conclusion

PSM Insight supports the PSM issue-driven measurement process by providing examples presented in the PSM Guide and allowing user-defined issues. Because PSM Insight's design is compliant with industry standards for Open Database Connectivity, PSM Insight can accept data from an existing software development process.

The Army's experience with previous metrics management tools and the use of COTS products expedited the development of PSM Insight.

PSM Insight can be obtained at no cost through the Army Software Metrics Office. To obtain a copy of the tool, or for more information on the issue-driven software measurement process, visit the Army Software Metrics Office Web site at www.ArmySoftwareMetrics.org or the PSM Web site at www.psmsc.com.

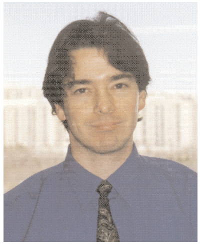
Note: The PSM process has served as the basis for ISO/IEC 15939, Software Measurement Process Framework. The PSM process may change with the creation of ISO/IEC 15939 to support the standards community. ISO/IEC 15939 may be more formally implemented in PSM Insight. Readers can access www.iso14001.net/iso15939/ to read the "Scope of the proposed ISO/IEC 15939 standard." ◆

Acknowledgments

I acknowledge Dave Morris of IEI for his efforts in developing PSM Insight. Jack McGarry managed the PSM project for years, and Cheryl Jones currently manages the project. More information may be obtained through the PSM Support Center:

U.S. Army
TACOM-ARDEC
AMSTA-AR-QAT-A, Building 62
Picatinny Arsenal, NJ 07806-5000
Voice: 973-724-5638
Fax: 973-724-2382
E-mail: psm@pica.army.mil
Internet: www.psmsc.com

About the Author



Don Scott Lucero is a software engineer on the headquarters staff of the Army's Operational Test and Evaluation Command (OPTEC). He is responsible for the Army's Software Metrics Office as well as OPTEC's software test and evaluation policy and methods. He has 15 years experience working on Army software development projects and has a bachelor's and a master's degree in computer science.

U.S. Army Software Metrics Office
Attn: CSTE-OM-SMO, Park Center IV
4501 Ford Avenue
Alexandria, VA 22302-1458
Voice: 703-681-3895
Fax: 703-681-6914
E-mail: lucerodon@hq.optec.army.mil

References

1. Office, Director of Information Systems for Command, Control, Communications, and Computers (DISC4, SAIS-ADW) Memorandum, "Acquisition Reform and Software Metrics," dated Sept. 19, 1996.
2. *Practical Software Measurement (PSM): A Foundation for Objective Project Management*, Version 3.1a, April 17, 1998.

It Is the People Who Count in Measurement: The Truth about Measurement Myths

Carol Dekkers, *Quality Plus Technologies, Inc.*
Mary Bradley, *MSB2*

The most overlooked aspect of software measurement is the effect on the people involved. The introduction of measurement in an organization involves a cultural change, but how much people affect the success of measurement is seldom anticipated or examined.

Here we debunk 10 of the most common management myths related to measurement and look at how people issues can ultimately cripple a measurement program or lead to its resounding success. We also talk about strategies based on our consulting experiences to help readers overcome obstacles in their organizations.

Myth No. 1:

People need only to know about the benefits, and measurement will sell itself.

Fact: Knowing the long-term benefits of measurement is important, but this is not enough to sell measurement in an organization.

Tell people the truth, repeatedly, about the benefits and the obstacles in implementing measurement—and it must be a consistent truth at all levels. Workers are insightful and know when the truth is being obscured.

It is critical that communication be consistent about measurement initiatives, particularly in the following areas.

- The reasons for measurement. Be honest and to the point. If measurement is intended to combat outsourcing threats or to find ways to reduce the work force, say so.
- Realistic time frames. It may take up to two years before the capture and analysis of data yields quantifiable results.
- Staff involvement. What is the expectation for each staff level for each phase of the measurement program? Will overtime be involved? How will their jobs change?
- What is in it for measurement participants? For example, increased estimating accuracy will provide management with realistic schedules and reduce unrealistic pressure on development staff to deliver before the software is finished.

Position measurement as a management tool for improvement, not as a "big brother" tactic to instill fear in the staff. Given the reduction mentality of the 1990s, skepticism and insecurity in information technology is rampant. With the introduction of any change, many people will hear what they want to hear, and some will think the worst. Management may be looking for reasons to outsource or cut back, and it is inevitable that those being managed will ask, "What is in it for me?"

Also, the rapidly changing world of technology leads many to believe that measurement is just another "program of the month" that will go away. Already under pressure to do more with fewer resources, many systems professionals believe time is wasted on activities that do not generate program code.

Another fear—measurement will show that "we are not as good as we have been saying we are or as good as the rest of our industry." It is more important, therefore, that participants receive some practical benefits of measurement, rather than merely being told about them.

Myth No. 2:

The right way to start a measurement program becomes apparent once you identify corporate objectives. Then, you just need to implement it.

Fact: Every measurement program should be aligned with the corporate objectives, but corporate objectives often do not include any "people" factors. Since people are the participants in a measurement program *and* the source of the data, poorly planned or haphazard attempts at measurement may compromise the program and reduce the anticipated benefits.

When planning your measurement program, consider the following to resolve some of the people issues.

- Upper management routinely develops the corporate objectives, but most of those in the organization do not know or understand those objectives. Ensure participants understand what the corporate objectives are and how measurement directly or indirectly links achieving those objectives. Everyone involved in measurement needs to know how their participation will benefit their position or the corporation as a whole.
- Individual corporate objectives often address a singular direction or desired marketplace position and exist to complement other corporate objectives. As such, one objective may indicate that a particular metric is important, without addressing other complementary metrics. One or two isolated measures will not be enough to build a sustainable metrics program.

A good program requires a balanced suite of measures that track achievement toward the critical corporate objectives. In this way, the gains toward one corporate objective are not to the detriment of another.

Myth No. 3:

There always will be people who resist change. Just give them time and they will come around.

Fact: Resisting change is more common than not, and a few hard-core resistors can sabotage an entire measurement program. Some think that anything new cannot be good or necessary. Others revel in the attention that being a detractor can attract. These people will only come around if they receive some benefit in the form of a tool they can use and understand. If they do, they likely will become the program's strongest supporters. It is well worth the extra effort to educate and work with these individuals to plan and implement measurement. Once they have been a part of the process and understand what is in it for them, former resistors often become your best lobbyists for measurement.

Myth No. 4:

Teach people the basics of measurement, then they will not need ongoing presentations.

Fact: Marketing professionals attest that the successful introduction of change relies heavily on frequent, effective presentations. People require many exposures before assimilating information, and measurement and its use are a complex subject.

In the first few exposures, people grasp the minimal information they need to get started. Providing only basic information yields only basic results on most projects. Newer technologies require advanced measurement techniques and better, in-depth use of data. Ongoing training keeps the measurement program in focus and on track, ensuring that changes are quickly disseminated to program participants.

Myth No. 5:

Software measurement is easy.

Fact: It is tempting to say that measurement will be easy and painless, to encourage participation. That is not always true. Such statements could damage the credibility of the entire program.

Software measurement is a complex subject that is pondered, discussed, and debated by some of the best software engineering minds in the world. Good training eases the usage of software measurement, but a few "casebook" systems and even the function-point rules that seem simple to understand are not always easy to apply.

In function-point counting, the counters need to know that even the experts sometimes have questions and that questions are preferable to producing invalid data. Measurement is a discipline that requires both effort and financial investment. There are no simple shortcuts to accurate measurement, but the journey can be rewarding to all involved.

Myth No. 6:

People can manage the start-up of the measurement program in addition to their current job.

Fact: It takes a full-time, dedicated resource to plan, do, check, analyze, report the results, and act on the results of a new software measurement program.

Although the budget cycle and budget constraints of many organizations can make this difficult, a successful, planned measurement initiative *does* require at least one full-time, dedicated resource. This resource also needs management's full support.

Myth No. 7:

Anyone who is available is a good candidate for the measurement coordinator.

Fact: Wrong! To properly introduce a cultural change, such as measurement, requires a change agent with knowledge of the subject and strong communication skills. Measurement programs rely heavily on marketing and require strong skills in human resources, data gathering, analysis, presentation, effective communication, and metrics. The measurement coordinator must balance the measurement program's needs with the measurement participants' readiness to accept and embrace change. This person can make or break the software measurement program.

In addition, the metrics person or function—even in a mature metrics organization—cannot be placed at random on an organizational chart. It is critical to the program's success to place the measurement function under the senior management who endorses and believes in the measurement initiative. More than one measurement program has suffered demise by reorganization.

Myth No. 8:

Measurement data brings its own rewards.

Fact: True to a limited extent. However, an important part of any new program's success lies in recognizing people and their accomplishments. Participants appreciate some acknowledgment of their efforts. Small tokens such as certificates, coffee mugs, ribbons, and thank-you's from management go a long way toward making people feel good about the program. Positive reinforcement of positive actions leads to even greater success.

Myth No. 9:

We will tell people about metrics on a need-to-know basis.

Fact: Corporations where this is a communication policy need to relax it when it comes to metrics and other corporate changes where success is dependent on people. Secrecy breeds notions of conspiracy, especially in an environment rife with downsizing, outsourcing, reengineering, or reorganization.

If your measurement program is truly to measure the process and not individuals, minimize rumors to the contrary by posting the minutes of your metrics meetings. Once the published information concurs with your presentations and with what management is saying, people become comfortable with the measurement initiative. Keeping minutes and plans secret fuels the rumor mill and churns out misinformation.

Remember, perception becomes reality in the absence of facts.

Myth No. 10:
Companies with outsourcing agreements that include measurement are naturals for measurement success.

Fact: Outsourcing arrangements bring their own unique set of people issues, regardless of whether measurement is involved.

Outsourcing agreement terms usually include only vague references to measurement, including what the measures are, how they should be used, which party should be responsible, and what the measurements mean. Often one of the first "projects" following outsourcing is to implement the measurement requirements outlined by the contract. Any measurement initiative can be sabotaged by people who say they understand measurement and its uses and do not.

Conclusion

These management myths are the root of many common people issues in software measurement. This list is not exhaustive. Other myths, such as "we are different; you can't measure us," prevail with development staff. Again, they involve people issues that require resolution. A separate article addressing these developers' myths is available from the authors.

Clear communication with and among people is the most important element in software measurement success. It is the people who count in measurement—recognizing and debunking the common management myths in your organization will take you far on the journey to measurement success. ◆

About the Authors

Carol Dekkers is president of Quality Plus Technologies, Inc., an industry-leading consulting firm specializing in software measurement and International Function Point Users Group (IFPUG)-certified function-point training and consulting as well as requirements, estimation, and process improvement consulting. She is president of the IFPUG Board of Directors and is a project editor within the International Organization for Standardization (ISO) functional size measurement work group (ISO/IEC/JTC1/SC7 WG12). She is a frequent presenter and trainer at both U.S. and international quality and measurement conferences and holds credentials as a certified management consultant, a certified function-point specialist, a professional engineer (Canada), and an information systems professional.

Quality Plus Technologies, Inc.
8430 Egret Lane
Seminole, FL 33776
Voice: 727-393-6048
Fax: 727-393-8732
E-mail: dekkers@qualityplustech.com
Internet: http://www.qualityplustech.com

Mary Bradley is a partner in MSB2, a software metrics and management consulting firm based in Mequon, Wis. She has extensive experience in management, training, quality, and productivity measurement in a broad range of industries. An IFPUG member since 1989 and one of the first certified function-point specialists, she has served on the Counting Practices Committee since 1994 and is currently chairwoman. She also is a member (former chairwoman) of the IFPUG's ISO Committee and a U.S. representative to the ISO software engineering work group on functional size measurement. She holds credentials as a certified quality analyst and a certified quality examiner.

MSB2
344 West Bonniwell Road
Mequon, Wis. 53097
Voice/Fax: 414-243-9130
E-mail: msb2@compuserve.com

USDA's National Finance Center: A 10-Month Journey to Level 2

A. Todd Steadman
TRW Avionics Systems Division

The National Finance Center (NFC), an agency of the U.S. Department of Agriculture (USDA), has been honored for excellence in information technology. Government Computer News *recognized NFC in March for its accomplishments in implementing the necessary software process improvements that are required to achieve Level 2 of the Software Engineering Institute's Capability Maturity Model (CMM) for software. NFC's efforts placed it among the top 30 percent of all federal organizations assessed under the CMM since 1986. This article describes the work that led to the Level 2 certification and provides insight into some of the successes, challenges, and lessons encountered along the way. The center is under the direction of John Ortego.*

Preparations

The NFC performs the record keeping and software development functions for the Federal Retirement Thrift Investment Board. Under Congress' direction, the board oversees the administration and operation of the Thrift Savings Plan, the 401K-like component of the federal retirement system. The NFC's CMM efforts were centered in its Thrift Savings Plan Division (TSPD) directed by Roderick Keith.

The board became interested in the CMM in early 1995. There was an initial assessment in August 1995, followed by an interim Software Capability Evaluation (SCE) in March 1997. TSPD enlisted the help of the Software Technology Support Center (STSC) later that year to assist with strategic planning, to develop and present software process improvement workshops, and to provide specific consulting in software process improvement and preparations for CMM assessments. Charles Stensrud, former NFC computer specialist program analyst, formed the TSPD Software Engineering Process Group (SEPG) and began to document the organization's software processes with respect to the CMM key process areas and identify improvement areas, training needs, and training sources.

A 10-Month Journey

NFC's journey commenced as efforts began in earnest in November 1997 with the organization of and subsequent rollout of several key processes. The processes included requirements management, project planning, tracking and oversight, and software quality assurance (SQA). In early 1998, the software configuration management (SCM) process was rolled out. Specific process action teams (PATs) researched the processes, which were developed and authorized by the SEPG and the Management Steering Group (MSG). The STSC presented a series of software project management workshops to personnel who were directly affected by and contributed to the new processes. The workshops were specifically tailored to the processes and included theoretical instruction in project management techniques, software tool use, and exercises in implementing the processes. The STSC continued to provide consulting services to TSPD, including document and process review, MSG support, analysis of data and results, and additional assessment preparation activities.

In March 1998, TSPD underwent a "CMM gap analysis" assessment. The results indicated a maturity level for the organization below Level 2 and that additional work was required prior to the external assessment planned for September 1998. Linda Giffin, NFC systems accountant, led several of the PATs and was responsible for the development of several of the key project planning, estimation, and management processes.

The gap analysis "was an extremely positive, extremely painful experience," Giffin remarked, but they "thought they had made it." She concluded that although the organization was extremely disappointed with the results, it seemed to energize their resolve and vigor to improve before the September assessment. TSPD continued to gather data from their processes and to improve and institutionalize them. In the first part of September, an external assessment was performed, and on Sept. 18, 1998, TSPD was officially certified CMM Level 2. NFC's future plans are to maintain the Level 2 rating in TSPD, transition the processes and success to the other divisions of NFC, and expand efforts toward Level 3 certification.

Roderick Keith has been reassigned and is director of the NFC's Application Systems Division (ASD) and has asked Giffin and Stensrud to lead the ASD effort to achieve CMM Level 2.

Challenges and Successes

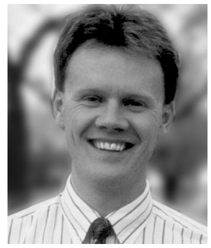
A number of challenges were encountered and overcome by TSPD in its journey to Level 2. Perhaps the strongest challenge was to manage the culture of the division in such a way as to improve the chances of success. As the CMM initiatives began, personal heroics were responsible for much of the division's success. There also was a strong sense of "things are always done this way," and a few skeptics seemed to always be present to predict the failure of any new idea. Another challenge was the time frame allotted to achieve Level 2. The board had directed that future funding for TSPD would be based on a successful Level 2 certification by September 1998.

TSPD responded to these challenges by gaining early management support for its CMM initiatives. TSPD established strong, opinionated champions throughout each level of management that ensured that the vigor of change would not be lost. The SEPG and MSG worked together to develop policies and processes and to strategically plan the steps toward successful achievement of Level 2. Skepticism was managed, successes were readily communicated, and feedback mechanisms were put in place, so everyone associated with the effort understood the plans, processes, and goals of the journey. The STSC's expertise was used to train personnel to operate in a Level 2 environment and to guide the MSG and SEPG in collecting and interpreting results and maintaining the course. In the end, the organization's culture had been managed, the changes had been successfully implemented, and the skeptics silenced—some of them becoming the most successful implementers of the new processes and ideas.

TSPD has some advice for those who find themselves struggling toward Level 2 and for those who may have just made it:

- Use SQA to your advantage. TSPD established an SQA team early on and appointed it to provide the monitoring function to meet the Level 2 goals.
- Keep management actively moving in the different processes. The effort must maintain its vigor to succeed, and management is uniquely qualified to ensure that it happens.
- Use both industry and the organization's expertise to its full extent. If needed, acquire any additional expertise needed to meet objectives. ◆

About the Author



A. Todd Steadman is a software development engineer for TRW Avionics Systems Division. He has provided software engineering technical services to the STSC for eight years. His efforts for the STSC have focused on software project management technology, including technology research, evaluation, and tool insertion in various Department of Defense and other federal government organizations. His contributions have included *CrossTalk* articles and project management and cost estimation technology reports produced and distributed by the STSC. He has a master's of science degree in computer science from Utah State University and a bachelor's degree in electrical engineering from the University of Utah.

TRW Avionics Systems Division
Ogden Avionics Engineering Center
1104 Country Hills Drive
Ogden, UT 84403
Voice: 801-625-8019
Fax: 801-625-8081
E-mail: todd.steadman@trw.com

FMSO Achieves CMM Level 4

The Navy Fleet Material Support Office (FMSO) has been rated a Capability Maturity Model Level 4 organization. They were assessed Oct. 5-9, 1998 under the CMM-Based Appraisal for Internal Process Improvement (CBA-IPI).

FMSO provides information technology products and services to the Navy, Department of Defense, and other federal activities, with particular emphasis on systems that support Naval supply, inventory and material management, financial processing, and maintenance operations.

FMSO is located on the Naval Inventory Control Point complex in Mechanicsburg, Pa. It serves as the Central Design Agency for software support in the Naval Supply Systems Command.

FMSO is the first Navy activity to achieve a CMM Level 4 rating. This certifies that FMSO has the ability, commitment, and established procedures to fully satisfy the CMM requirements of a Level 4 organization and is performing based on FMSO's defined processes.

FMSO has gone from a work force of more than 1,410 employees in 1990 to a little more than 880 in 1998. Faced with personnel reductions and retirements, the organization had to capture the "corporate knowledge base" and learn to capitalize on the organization's strengths. Top-level management support helped improve areas of weakness.

The first assessment (October 1992) yielded a CMM Level 1 rating. Three years later, FMSO achieved a Level 3 rating. As a CMM Level 4 organization, they look forward to the challenge of expanding their use of new technology in software development, while stepping up to attain a Level 5 on the next assessment.

FMSO's goal is to provide the integrated solutions necessary to support the complex business changes its customers need. FMSO's success is directly related to its ability to perform predictably, yet do more with less.

The assessment team consisted of Kathy Chastain, Joe Bobby, Ron Doyle, Skip McGowan, Deb Yorlets, and Dave Shupe of FMSO, along with John Smith, Ann Roberts, and Felecia Hensley of Dayton Aerospace, Inc.

Navy Fleet Material Support Office
5450 Carlisle Pike, P.O. Box 2010
Mechanicsburg, PA 17055-0792

Software Process Improvement Department
(Code 902)
Attn: Kathleen D. Chastain
Voice: 717-605-2925 DSN 430-2925
Fax: 717-605-8213
Internet: http://www.fmso.navy.mil/index.html

Mission-Critical and Mission-Support Software: A Preliminary Maintenance Characterization

Elizabeth K. Bailey Clark, *Software Metrics, Inc.*
James A. Forbes, *Logistics Management Institute*
Emanuel R. Baker, *Software Engineering Consultants, Inc.*
Donald W. Hutcheson, *Weapons Business Development, The Boeing Company*

Within the Department of Defense (DoD), mission-critical software maintenance has been reported to cost between $700 million and $20 billion annually. The wide range of estimates results from uncertainty over the definitions of "mission-critical" and "software maintenance" as well as the lack of any catalog of performing activities. The problem is deeper than definitions and level of investment: software maintenance process is poorly characterized in general. The purposes of this study[1] were to undertake an initial characterization of DoD mission-critical software maintenance in terms of its activities and processes, users and stakeholders, amount of resources, and existing formal and informal policy; to identify policy issues; and to outline the scope and major features of potential new or revised policy.

THE PROBLEM, HOWEVER, is deeper than definitions and level of investment.

The software maintenance process is poorly characterized in general. Lacking an adequate characterization of software maintenance, there is no real basis to establish coherent policy. Further, key software maintenance decisions—such as contract choice or organic performance, and whether it should be defined as depot maintenance—are largely ad hoc and reap limited benefits from the results of past decisions.

The terms "software maintenance" and "software support" are both in use, sometimes with modifiers such as "post-production" or "post-deployment." To avoid confusion, we adopted the term software maintenance and defined it to include

- Correction of defects.
- Adaptation to a new host operating environment.
- Incremental functional improvements.

This definition is generally consistent with industry usage. Excluded from this definition are major modifications and upgrades, the purpose of which is major functional improvement.

We found it helpful to distinguish among three categories of mission-related software: mission-critical, embedded; mission-critical, nonembedded; and mission-support (Table 1). Broadly speaking, different organizations may use similar processes within a category; across categories they generally do not.

It also is helpful to characterize software maintenance by application area (Table 2). We gathered data on the first six application areas of Table 2 (shaded in the right column). Given the state of data availability and reasonable limits on study scope, it proved impractical to assure completeness for any category or to achieve a reasonable degree of completeness for other than the first three.

Approach

Our study approach is illustrated in Figure 1.

We separated the research into two segments: quantitative and qualitative. To establish the "demographics" of software maintenance, e.g., rough order of magnitude estimates of the code base, number of people performing, and annual cost, we started with a database created by the Institute for Defense Analyses for the Commission on Roles and Missions (CORM) of the armed forces. Because it was clear from the beginning that this database (the result of a data call to the services) had some voids, we supplemented it with data we obtained directly from the services. This study does not include software maintenance performed by defense agencies; the decision to exclude defense agencies was driven by the need to establish a reasonable scope of effort for what was envisioned as primarily an exploratory study.

To approach the more qualitative aspects, such as those that have to do with the software maintenance process, we began with a literature review and conducted a series of 15 semistructured interviews at eight service installations. In keeping with the unsettled nature of software maintenance, we focused on developing an understanding of the common norms, meanings, values, and organizational relationships [1]. We were more

Table 1. *Software maintenance categories.*

Type	Cardinal Characteristics	Examples
Embedded	• Tightly coupled interfaces • Real-time response requirements • High reliability requirements (life-critical) • Generally severe memory and throughput constraints • Often executes on special-purpose hardware	B-1 flight software, F-14 flight software
Operational, nonembedded	• Multiple interfaces with other systems • Constrained response time requirement • High reliability but not life-critical • Executed generally on commercial-off-the-shelf products (COTS)	C^3, space systems
Mission-support	• Relatively less complex • Self-contained or few interfaces • Less stringent reliability requirement	automatic test, equipment test, program sets, mission planning, business systems

Application Area	Type	Data Completeness
Weapon systems	Embedded	Essentially complete
Space control	Nonembedded	Essentially complete
Automated test equipment	Support	Essentially complete
C3	Nonembedded	Partial
System integration labs	Nonembedded	Partial
Simulation and training	Nonembedded	Partial
Atmospheric search	Nonembedded	none
Wargames and mission rehearsal	Nonembedded	none
Intelligence	Nonembedded	none
Business systems	Support	none
Weather	Nonembedded	none
Other	–	

Table 2. *Scope of DoD software maintenance.*

interested in discerning signposts and perspectives[2] than trying to determine "facts." In combination, the demographics research, literature review, and interviews permitted us to do this by characterizing software maintenance in terms of activities and processes, users and stakeholders, amount of effort, and existing formal and informal policy. Policy issues flow from that characterization.

Findings [2]

Within the scope of the study, we accounted for an estimated 16,000 government and contract persons performing software maintenance on 278 million source lines of code (SLOC) at a cost of $1.26 billion annually. We found that approximately 55 percent of these people were government employees, and 45 percent were contractors. Approximately 40 percent focused on software correction, and 60 percent focused on a combination of adaptive and incremental improvements.

Code Base

Figure 2 shows a breakout by the three high-level categories for each service. The Navy and the Air Force have much larger code bases than does the Army.

Although support software is the single largest category in terms of the sheer number of SLOC, it is less costly to maintain than the other two categories. As an indicator of the difference, Table 3 reflects the approximate cost per SLOC per year for three of the sites in the expanded database.

In interpreting Figure 2, remember that there are significant reliability and validity issues with the underlying data. Although our check of code counts reported in the CORM database against those made available in site visits did not reveal a systematic bias, that is not the same as saying the data are known to be valid. Because only three of the six application areas we examined were reasonably complete, this summary is an underestimate even for the areas we examined. The portrayals shown here are best characterized as approximate representations of the relative sizes of the code bases for the categories we examined. These caveats also apply to the labor force demographics presented and budget impact.

Personnel

Use of operations and maintenance (O&M) funds is almost universal for software maintenance within the application areas studied. The amount of resources is normally determined as a level of effort rather than built up from discrete requirements. In some organizations, the level of effort was fixed in terms of dollars, in others by the fairly stable size of the labor force. In either case, software maintainers addressed the backlog of requirements to the extent resources permitted. Requirements not satisfied in one planning period, e.g., year, were deferred to the following period. This approach also appears to be consistent with industry software maintenance practice.

Software development and maintenance are labor-intensive. Human effort is generally recognized to be the major cost driver [3, 4]. To estimate the number of people involved in software maintenance, we began with the CORM database personnel counts. Here also, we expanded the CORM database using other data gathered during the study. To determine accuracy, we compared, as we did with the size data, the numbers obtained from the site visits with those in the CORM database.

The CORM database consistently underrepresented the number of people. A comparison between the CORM and the site visits is shown in Figure 3. If the data from the site visits and the CORM data for the same sites were about the same, a

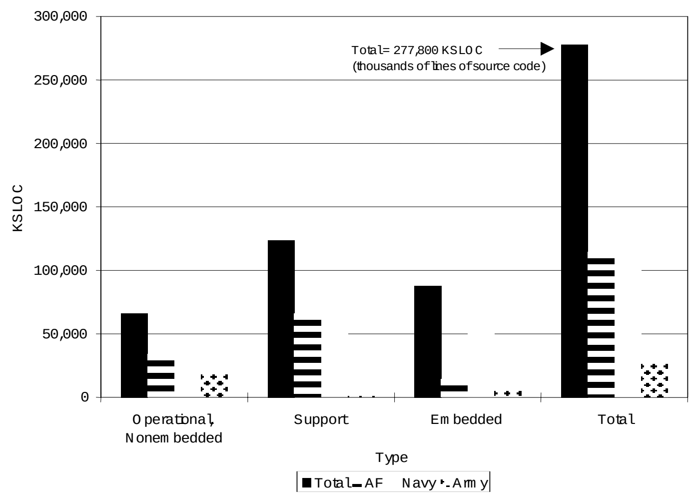
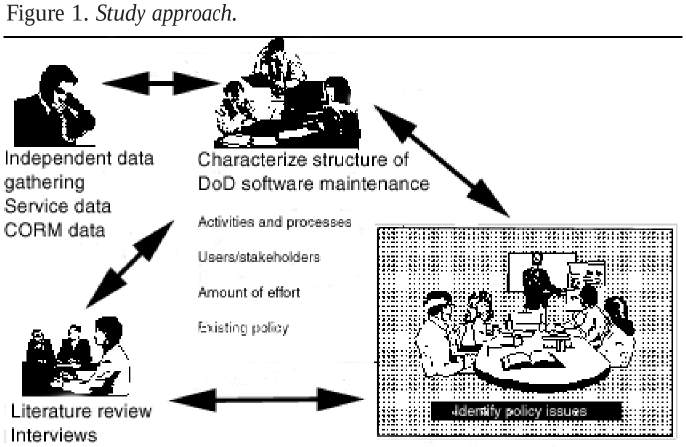


Figure 2. *Software code base by service and category.*

Figure 1. *Study approach.*



Category	Approximate maintenance cost per line of code per year
Embedded	$110.00
Nonembedded	$5.60
Mission-support	$0.81

Note: The mission-support cost is calculated from North Island ATE TPSs, nonembedded is calculated from CECOM data, and embedded is calculated from B-1B data.

Table 3. *Representative maintenance costs by category.*

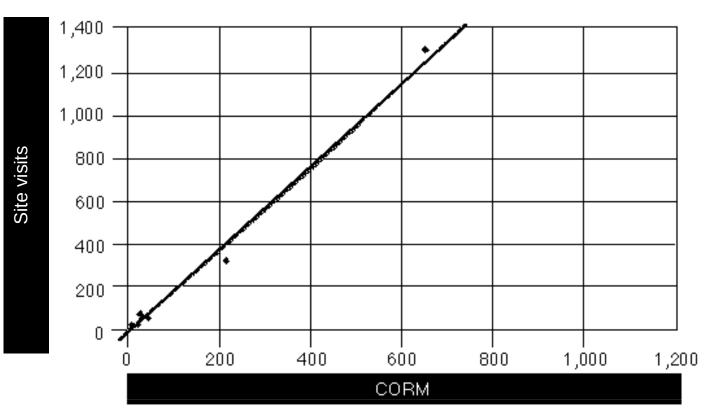
linear plot of the data would have a ratio of 1-to-1 slope. The slope is 1.96, which means that the personnel counts obtained from the site visits were almost twice as large as those from the CORM data call, and this was consistent for all but one of the sites we visited. The one inconsistency was the F/A-18 Hornet aircraft. The CORM data call reflects 30 F/A-18 personnel, all organic, while interviews with F/A-18 software managers indicate the total should be approximately 1,000 (125 organic plus 875 contractors). Since it was such an egregious error, we did not include the F/A-18 in calculating the 1.96-to-1 site-visit-to-CORM data ratio.

Budget Impact

The third measure of magnitude is dollars. We did not use the budget numbers from the CORM data call because it is unclear what these reflect, i.e., labor only or labor and equipment or contract or contract plus organic. As an alternative, we estimated the financial commitment in dollars by multiplying counts of people by average loaded labor rates for organic and contractor personnel. Figure 4 shows the estimated dollars per year for each service.

The rate used for organic personnel was $67,364, which is a composite rate based on an assumed distribution of 80 percent GS-12 and 20 percent GS-13 (1996 dollars) [5]. The rate used for contractor personnel was $97,364, which is the median of the rates that were quoted to us during the site visits. The contractor rates ranged from $55,500 to $250,000 per year, and this difference generally corresponded with the complexity and

Figure 3. *Personnel data from eight site visits compared to CORM data for same sites.*



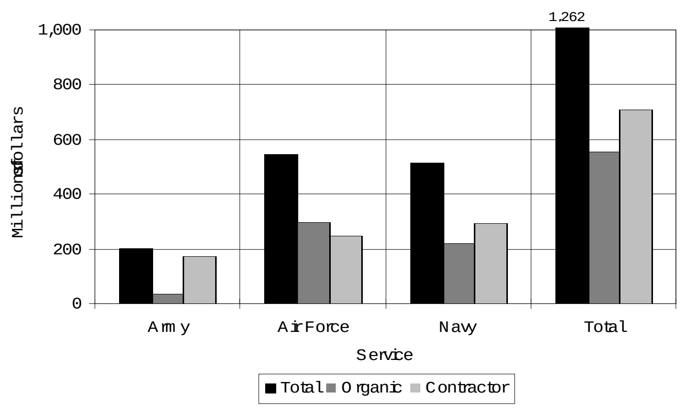
uniqueness of the software being maintained. The difference between organic and contractor rates should not be interpreted to mean that contractors are more expensive. By and large, the contractor labor force was maintaining more complex software that required higher skills. More to the point, we did not attempt to make such a comparison.

The financial commitment that we were able to account for using this procedure is approximately $1.26 billion annually ($205 million for the Army, $543 million for the Air Force, and $514 million for the Navy).

One of the reasons for characterizing DoD software maintenance was to shed light on the amount of software maintenance that also is depot-level maintenance. It is of interest whether software maintenance is depot level because it affects the department's compliance with the congressional restrictions on how much depot maintenance work can be outsourced [6].

It was not possible to describe what fraction of the $1.26

Figure 4. *Estimated budget impact by service.*



billion in software maintenance is depot level. First, it was clear from the interviews that, here also, there is a lack of consensus over definitions. For example, the Air Force would generally classify work on fighter aircraft embedded software as depot maintenance. The Navy did not consider it so. Hence, inclusion or exclusion of software maintenance when reporting compliance with Title 10 U.S.C. limitations on depot maintenance outsourcing was inconsistent. There was a lot of uncertainty in this area, as were differences in counting rules. The Defense Depot Maintenance Council Business Plan for fiscal 1996-2001, which is compiled with service inputs, showed $275.3 million in contract depot-level software maintenance for fiscal 1996 and an additional 3.2 million depot labor hours of organic support. By contrast, the AP-MP(A)-1397 Depot Maintenance Cost System Report, under which depot-level software maintenance was explicitly required to be reported, reflected $20.4 million for the same year.

Transition Patterns

Software for the application areas studied normally is developed in the private sector. Although there were many transition patterns from original equipment manufacturer (OEM) to maintainer, three reasonably clear trends emerged:

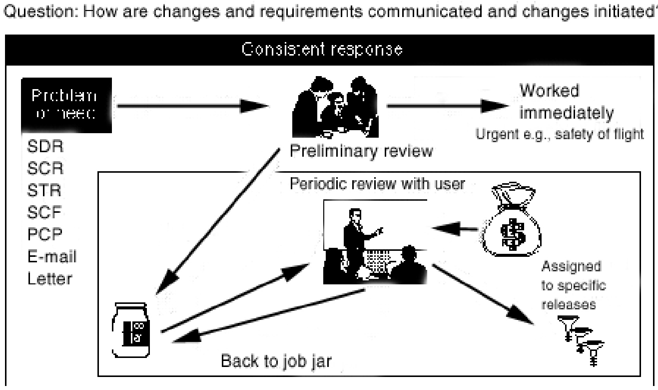
- Pure organic maintenance is the exception and seems limited to mission-support software, such as ATE TPSs. Since the organic and contract sectors have roughly the same skills and would be expected to use the same software environments, we conclude that, except for support software such as ATE TPSs, there is significant difficulty and cost associated with transferring the knowledge of the software necessary for its maintenance. In addition to problems with nondelivery of documentation or computer-aided software engineering environments, this knowledge is probably tacit, i.e., deep knowledge, rather than explicit—that is what makes it hard to transfer. What we have found in practice seems to support this conclusion. Support for this conclusion also is found in the literature on technology management in which David J. Teece [7], examining how companies arrive at make-or-buy decisions, noted that they often choose what is easy to do rather than what is most important to them.
- Organic maintenance of embedded software generally is found only on older models of weapons systems.
- Where attempted, competitive contract support proved both more economical and at least as effective as either sole-source contract support or organic support.
- Based on the empirical evidence, i.e., the established transition patterns, planning for pure organic maintenance or competed maintenance of embedded software is unrealistic. It probably is more realistic to accept OEM involvement in (and initial lead of) embedded software maintenance as an accomplished fact.
- Competed commercial maintenance is viable for mission-critical, nonembedded and for mission-critical, support software.

Communication of Requirements

Communicating requirements clearly is an important part of the software maintenance process. We found uniformity in this process among organizations in the field survey. The typical requirements process (Figure 5) follows these steps:

Figure 5. *Requirements process.*

Question: How are changes and requirements communicated and changes initiated?



Note: SDR = Software deficiency report; SCR = Software change request; STR = Software trouble report; SCF = Software change form; PCP = Program change proposal; E-mail = Electronic mail.

- A user initiates it through a problem report or a change request. These reports or requests had almost as many names and acronyms as organizations surveyed. The names included System Deficiency Report, Standard Change Form, Software Trouble Report, and Program Change Proposal, or they could take the form of E-mail or letter input. Interestingly, no one in the Air Force reported using formal Technical Order 00-35-D54 deficiency reports, though this technical order applies to all Air Force agencies and organizations and provides for software deficiency reporting [8].
- The requests typically are screened in a preliminary review to determine the urgency of the problem or change request. Urgent needs, e.g., safety of flight, are worked immediately. The remainder of the requests are accumulated in what the Space and Warning Systems Directorate colloquially termed a *job jar* awaiting a scheduled review [9].
- The requests are periodically reviewed by an established group, e.g., F/A-18 System Change Review Board. Prior to the review, initial estimates of the magnitude of the effort—which changes can be efficiently grouped, etc.—are accomplished by an engineering staff. The reviews often have user participation or input. The group chartered to do the review examines the requests in the job jar, prioritizes them, and selects software changes to be implemented. Selection is based primarily on priority and available funding.
- Requests not selected go back to the job jar for future consideration. Typically, there are more requests than funds.
- Problem reports or change requests selected for implementation are assigned to a software version release.

Neither the size of the backlog of requirements nor the specifics of particular requirements in the backlog drives the budget. Rather, planned support takes the form of a level of effort expressed in dollars or work force. Essentially, the agreed-upon level of effort establishes a "cut line." On a prioritized list of software maintenance requirements, software changes above the line are implemented; those below it are deferred to the job jar for future funding opportunities. This behavior would indicate that most software maintenance tasks are not of a time-critical nature. It is worth noting that level-of-effort funding is found in commercial software maintenance practices [10]. (There are at least anecdotal indications that it also is found in commercial software development.)

Operable Policy and Military Standards

A primary reason for this study was to understand what is needed in the area of software policy. Consequently, this topic was explored in some detail during the interviews. Policy can be viewed from two different perspectives. First, it can be considered as representing required behavior, i.e., as formal, normative policy, or the common view. Another perspective is to consider policy as providing a framework of consistent expectations regarding how affected parties mutually interact, i.e., as facilitating cooperative action [11,12,13,14]. Given the relative absence of normative software maintenance policy, both perspectives were potentially important.

The most frequently cited documents were several military standards that prescribed software engineering processes. Almost universally, DOD-STD-2167 or DOD-STD-2167A were mentioned. Several respondents listed MIL-STD-498 as well. Two sites mentioned MIL-STD-1679.[3] These military standards describe the documentation to be delivered, formal reviews to be held, and tasks to be addressed in developing or maintaining software. A fairly broad variety of other documents also were listed. These included DoD (especially 5000 series), service, and command regulations and instructions.

It was clear that military standards are the most important source of policy for software maintenance. The single most important reason for this was that the military standards provide a consistent framework of expectations for software developers and software maintainers—two communities that generally have limited interaction during software development. It is on the basis of what is described in the military standards that the software maintenance community knows what to expect in the way of software documentation. A considerable unease was expressed in almost all of the interviews regarding the demise of the military standards. This unease stems from the potential loss of this consistency of expectation. One expectation was the Navy's F/A-18 program, which has successfully eliminated the wall between developer and maintainer through the successful use of integrated product teams (IPTs) [15, 16].

Not surprisingly, given the de facto status of the MIL-STDs as policy, the ongoing elimination of MIL-STDs was an issue for almost all of the organizations we interviewed.

Recommendations

We made two sets of recommendations: one set related to general policy, and a second related to how DoD organizes for software maintenance.

Policy

- Standardize the term *software maintenance* and define it to include correction of defects, adaptation, and incremental improvements. Exclude major modifications.
- Define software maintenance in weapons systems, automatic test equipment, systems integration laboratories, and space control categories as depot maintenance. All four categories are either embedded in or closely tied to mission-essential platforms.
- Make routine the consistent reporting of depot-level software maintenance, as defined above, in the AP-MP(A)-1397 Depot Maintenance Cost System to provide a basis for reporting to Congress and management of depot-level software maintenance generally.
- Invest in process improvement. Consider mandating minimum process capability levels for both organic and contract activities that perform software maintenance.

Organizing for Software Maintenance

- To achieve scale economies, consolidate smaller software maintenance activities into software maintenance centers of excellence. For each center of excellence, keep or put in place a strong central management structure.
- For embedded software, plan for long-term OEM maintenance. However, it is important to retain enough work organically to maintain smart-buyer capability.
- For mission-critical, nonembedded software, continue consolidation using the government-managed, contractor-performed, centralized-maintenance model employed by the Army Communications Electronics Command and the Air Force Space Systems Support Group.
- For software, such as automated test equipment, test program sets where the software engineering knowledge is relatively easy to transfer. Consider competition to reduce costs. ◆

About the Authors

Elizabeth K. Bailey Clark co-founded Software Metrics, Inc. in 1983. She is a primary contributor to Practical Software Measurement (PSM) and is a qualified PSM instructor. She has worked with numerous clients to implement software measurement. She was a primary contributor to the Software Engineering Institute's core measures. She is currently working with Barry Boehm and Chris Abts from the University of Southern California to develop and calibrate COCOTS, a software COTS integration cost model. She has a bachelor's degree from Stanford University and holds a doctorate from the University of California at Berkeley.

Software Metrics, Inc.
4345 High Ridge Road
Haymarket, VA 20169
Voice: 703-754-0115
E-mail: bkbailey@erols.com

James A. Forbes is a certified professional logistician with more than 25 years of experience. As an Air Force officer, he had extensive experience assessing the impact of technology on organizations and managing technology-driven change. In the private sector, he managed logistics programs for both Technology Applications, Inc. and Analytic Sciences, Inc. before coming to the Logistics Management Institute (LMI). At LMI, he has been a principal researcher in studies of depot maintenance, total ownership cost, and logistics-related information systems.

E-mail: jforbes@lmi.org

Emanuel R. Baker is president of Software Engineering Consultants, Inc., Los Angeles, Calif., and a principal officer of Process Strategies, Inc., a software consulting firm based in Los Angeles and Walpole, Maine. The author or co-author of several articles on software quality, software quality management, software engineering, and software process improvement, he has 40 years of engineering experience, 25 of which has been spent as a software engineer and as a consultant to software development organizations. He held a number of management positions and was manager of the Product Assurance Department of Logi-con's Strategic and Information Systems Division. He has played a

major role in the development of software quality standards for both industry and the Department of Defense. He has a bachelor's degree in engineering (1956) from New York University, New York City, a master's degree in mechanical engineering (1958), and a master's (1974) and a doctorate (1981) in education from the University of Southern California at Los Angeles.

E-mail: erbaker@ix.netcom.com

Donald W. Hutcheson has extensive experience in engineering development, flight testing, program management, senior technology management, and executive management of DoD weapons systems. He joined LMI in 1994 as a logistics analyst. He has a bachelor's degree in mechanical engineering and a master's degree in engineering. He is a member of the DoD Acquisition Corps. certified in program management. While at LMI, he has performed analysis in various areas of logistics reform, co-wrote three reports, and was a recipient of the LMI 1996 President's Award.

Weapons Business Development
The Boeing Company
P.O. Box 516 MC S500-2025
St. Louis, MO 63166-0516
Voice: 314-925-5633
Fax: 314-925-7758
E-mail: donald.hutcheson@mw.boeing.com

References

1. Lyytinen, Kalle J. and K. Klein Heintz, "The Critical Theory of Jurgen Habermas as a Means for a Theory of Information Systems," *Research Methods in Information Systems*, E. Mumfored, et al., ed., Holland: Elsevier Science Publishers B.V., 1985, p. 221.
2. Kelle, Udo, "Theory Building in Qualitative Research and Computer Programs for the Management of Textual Data," *Sociological Research Online*, Vol. 2, No. 2, http://www.socresearchonline.org.uk/socresonline/2/2/1.html. 3.9.
3. Boehm, Barry W., *Software Engineering Economics*, Prentice-Hall, 1981.
4. Goethert, Wolfhart B., Elizabeth K. Bailey, and Mary B. Busby, *Software Effort and Schedule Measurement: A Framework for Counting Staff-Hours and Reporting Schedule Information*, CMU/SEI-92-TR-21, ESC-TR-92-021, Carnegie Mellon University, Software Engineering Institute, Pittsburgh, Pa., September 1992.
5. The composite organic rate is a weighted average of the rates shown for GS-12s and GS-13s in Table A26-1 of the *Civilian Standard Composite Pay Rates by Grade*, Air Force Instruction 65-503, May 1996.
6. 10 U.S.C. 2466, "Limitations on the Performance of Depot-Level Maintenance of Material," requires that not more than 50 percent of the funds available in a fiscal year to a military department or agency for depot-level maintenance and repair may be used to contract for performance by non-federal-government personnel.
7. Teece, David J., "Technological Change and the Nature of the Firm," *Technological Change and Economic Theory*, 1988, pp. 256-281.
8. U. S. Air Force, TO 00-35D-54, USAF *Deficiency Reporting and Investigation System*, Jan. 15, 1994, pp. 1-1, 1-4, and 1-5.
9. Space and Warning Systems Directorate, Operating Instruction 33-7, *Software Maintenance – Acronyms and Terms*, Vol. 2, Sept. 15, 1995, p. 6. The term *job jar* is used in this operating instruction.
10. Abran, Alain and Hong Nguyenkim, "Analysis of Maintenance Work Categories Through Work Measurement," *Proceedings of the 1991 IEEE Conference on Software Maintenance*, Sorrento, Italy, p. 105.
11. Forester, J., "Selling You the Brooklyn Bridge and Ideology," *Theory in Society*, September 1981, p. 746.
12. Forester, J., "The Policy Analysis-Critical Theory Affair: Wildavsky and Habermas as Bedfellow?" *Journal of Public Policy*, No. 2, 1982, p. 151.
13. Habermas, J., *The Theory of Communicative Action*, 1984, Vol. 1, p. 308.
14. Seidman, S., ed., Jurgen Habermas on Society and Politics: A Reader, Beacon Press, Boston, 1989, p. 154.
15. Bailey, Elizabeth K. and Beth Springsteen, The F/A-18E/F: An Integrated Product Team (IPT) Case Study, Institute for Defense Analyses, Alexandria, Va., April 9, 1998.
16. Springsteen, Beth, Elizabeth K. Bailey, Sarah H. Nash, and James P. Woolsey, Integrated Product and Process Development Case Study: Development of the F/A-18E/F (Draft Final), Institute for Defense Analyses, Alexandria, Va., April 22, 1999.

Notes

1. The study was performed pursuant to Department of Defense Contract DASW01-95-C-0019.
2. The findings presented are extracted from Logistics Management Institute Report LG518T1, November 1997 and represent the more significant findings of the study. For more detailed findings, please examine the report.
3. MIL-STD-498 replaced both DOD-STD-2167A (for weapons systems and other mission-critical applications) and DOD-STD-7935A (for automated information systems) and brought these two areas together under one standard.

PAIR: A Rational Approach to Fighting Software Project Fires

Gregory T. Daich
Software Technology Support Center

What happens when an organization finds a significant number of critical defects during testing? Whether the organization has a documented process or not, the organization naturally reverts to firefighting. It must quickly correct the problems before the schedule is "burned" beyond repair. The defined process for firefighting—which is likely called by other names, such as debugging, root cause analysis, or redesign or rework—may be followed. Regardless, it is project firefighting.

Red Adair

The book, *An American Hero: The Red Adair Story, An Authorized Biography* by Philip Singerman, tells of the famed firefighter who battles oil fires around the world. Revered as one of the most heroic firefighters, Adair has inspired us to assist organizations with immature document review practices to put out project fires. We have even seen some organizations with documented processes get caught firefighting too late in the project to overcome poor software quality.

PAIR Service

The Software Technology Support Center's (STSC) Preliminary Analysis Inspection Report (PAIR) Service has demonstrated to many organizations an effective document review (inspection) process.[1] This service identified significant opportunities for improvement as well as the cause of the fire raging through many projects. We review a sampling of the organization's project documentation that is causing problems. The service then provides a report that can be used to initiate and plan improvements that focus on achieving desired quality levels.

Typical Review Practices

After encountering a large number of defects, some people ask why the defects were not found in previous reviews. Document reviewers often are not given useful guidance in how to review documents. A manager merely slaps down a 100-page or more document and says, "We are going to have a meeting on this document in two days. I want you to review it."

The reviewer quickly skims through the document for the obvious problems or reads it late into the night to prepare for the meeting. Either way, many critical defects that could ignite project wildfires anytime during development are often missed.

Management usually has no problem signing off bug-infested documentation because project personnel cannot see the problems. More specifically, project personnel have not taken the time nor have they used effective techniques to inspect project documentation. It is not uncommon for organizations starting a new document inspection program to encounter 10 or more major defects per page in requirements specifications, designs, test plans, and process documentation.

Enlightened Review Practices

A simple set of document rules and other useful tools and practices can turn a document skimmer or near comatose reviewer into an effective consultant who advises document authors about significant document issues. The PAIR concept initially joins the project organization with the STSC as partners in fighting fires caused by serious document defects. The document review practices we advocate help organizations dramatically improve their ability to find and remove many types of serious defects when it is cost-effective to do so, and ultimately to prevent them from occurring in the first place.

Are CMM[2] and the J-STD-016[3] Enough?

Using the right tools, it does not take much effort to determine that a project has document quality problems. Many organizations have conducted process improvement initiatives, which started with a Capability Maturity Model-flavored process assessment. However, the guidelines for conducting these assessments are mainly concerned about document existence and not the quality of the project documentation. Some process assessors have reviewed a few document samples during an assessment and found useful information about process maturity.

I have seen approved software test plans that are void of test planning information. I have even seen software development plans that did not contain schedule, task, and product deliverable information. The main reason these significant problems exist is often because people do not fully understand the purpose of the required documents. Standards like the J-STD-016 should help us write better documents, but without effective and efficient review practices, the standards and guidelines do little.

Institute of Electrical and Electronics Engineers (IEEE) 1028, Software Reviews, provides some useful generic review practices that are worth considering. But this standard will still need to be customized for an organization's specific needs.

PAIR Service Demonstrated

Projects that are experiencing significant quality and testing problems can be assessed to determine if document quality is significantly inhibiting software quality and testing effectiveness. Understanding the purpose and objectives each document is supposed to accomplish and mapping the contents against those objectives often finds gaping holes in the document's content. A no-cost demonstration of our PAIR service has substantiated this fact for many organizations. It is a simple yet powerful approach to reviewing that changes how people look at documents and can radically improve overall project performance and software quality.

We will use our PAIR Service to help you extinguish fires that threaten your projects. Recovering from poorly written requirements documents during systems testing will be expensive compared to recovering during the requirements phase. However, it will be much less expensive than recovering after system delivery. PAIR will help you start a document quality improvement initiative before the fires rage and help you achieve your project goals.

Conclusion

We are not advocating firefighting as a way of life for software developers. We advocate a rational approach to stamp out fires when they occur. A consequence of this approach is the ability to prevent project fires and many types of software quality problems in the first place.

We do not need heroes who remove defects just before delivery after they inserted them throughout development. Red Adair does not start the oil fires he is asked to extinguish, but we need heroes like him in the software world who can put them out before we lose everything. We need heroes to identify significant issues early, helping authors improve document quality and succeed in meeting document and project objectives. ◆

About the Author

Gregory T. Daich is a senior software engineer with Science Applications International Corporation currently under contract with the Software Technology Support Center. He supports STSC's Software Quality and Test Group with more than 22 years experience in developing and testing software. He has taught more than 60 software test, document inspection, and process improvement seminars in the last five years.

Daich consults with government and commercial organizations on improving the effectiveness and efficiency of software quality practices. His consulting approach coordinates formal document inspections with analysis of test work products to identify opportunities for software test process improvement. These practices have also been applied in supporting and testing year 2000 upgrades. He is the principal author for several guidebooks and workshops for conducting year 2000 compliance projects. These guidebooks address corporate- and project-level compliance efforts as well as year 2000 desktop (PC and Macintosh) software compliance.

Daich has a master's degree in computer science from the University of Utah and a bachelor's degree in mathematics from Weber State University.

Software Technology Support Center
7278 Fourth Street
Hill AFB, UT 84056-5205
Voice: 801-777-7172
Fax: 801-777-8069
E-mail: daichg@software.hill.af.mil
Internet: http://www.saic.com

Notes

1. For example, see Tom Gilb's book *Software Inspections*, Addison-Wesley, 1993.
2. CMM stands for the Capability Maturity Model developed by the Software Engineering Institute with the support of many government and commercial organizations.
3. J-STD-016 is the Electronic Industries Association Institute of Electrical and Electronics Engineers Standard for Information Technology Software Life Cycle Processes Software Development Acquirer-Supplier Agreement, 1995.

Design Maturity Model

Donald Przebowski
Decisive Analytics

Until the invention of the Design Maturity Model (DMM), no existing models or procedures automatically linked and integrated the industry standards associated with metrics measurements, ISO 9000, and organizational responsibilities. Existing procedures had to be enhanced, not replaced, for the industry to accept the DMM. Cost for implementing the DMM had to be minimal, the DMM had to improve organizational discipline, and it had to facilitate development control. The final objective was to submit the DMM to a reputable organization for review and possible implementation.

Analyses of the Current Procedures

The initial analysis revealed that the software industry emphasizes metrics measurements to control software development. The Software Engineering Institute Capability Maturity Model (CMM), the Navy's Practical Software Measurements (PSM), the Army's Metrics Measurements, and recently ISO 9000 have evolved into industry standards. In general terms, ISO 9000 standards stress management and organizational responsibility and quality control. The standards are intentionally vague to permit organizations the freedom to create effective management policies. Eventually, it became clear that ISO 9000 standards can and should be reconciled to metrics measurements. The metrics measurements may be categorized into documentation control and performance measurement.

The metrics measurements used for documentation control are requirements traceability and design stability. Ordinarily, there are five main documents used to control software development. They are the requirements (the shall statements in the Statement of Work [SOW]), the specifications (design requirements), the functional description (the expanded requirements and the functions the system is expected to produce for the customer), the system interfaces (interactions between subsystems and systems), and the test requirements (test scenarios) to validate the system. The performance measurement standards are

- Breadth of Test – measuring the requirements tested, passed testing, and failed testing.
- Depth of Test – measuring the number of paths and conditions tested, passed testing, and failed testing.
- Complexity – measuring the degree of complexity, e.g., the number of lines of code.
- Computer Resources – measuring the storage capacity used by the system.
- System Response Time – measuring the time required for the system to respond to various actions.
- Defects/Faults – measuring the number of errors detected during testing.
- Lines of Code – measuring productivity, size, and complexity.
- Earned Value – measuring the variances associated with cost and schedule.

Insofar as documentation control, the analysis revealed that there was a misunderstanding of each type of document and purpose. For example, if one requested the SOW, one could be given the functional description or even the specifications. If one approached a tester for the SOW, one could be given the test document. Although this misinterpretation does not appear serious, an effective organization requires that every member understand the documentation if that member is expected to contribute to documentation control.

The analysis of performance measurement techniques revealed additional concerns. For example, during testing, the tester focused only on the test requirements document. If there was a design change in the requirements or the specifications that was not incorporated into the test requirements document, the tester would be unaware of that design change and, undoubtedly, test to obsolete requirements. There was no mechanism that automatically linked each type of document or a document's paragraph to the design requirements or specifications of a software item: computer software unit (CSU), computer software component (CSC), or corresponding computer software configuration item (CSCI). In addition, for a specific CSU, the requirements paragraph could be labeled 1111; the functional description's paragraph 2222, etc. One should consider the overwhelming task of first finding the appropriate paragraph, then analyzing and reconciling the design requirements to the test requirements line by line or paragraph by paragraph when the system contains multiple CSUs and a million lines of code.

Additional analysis revealed that the software organizations, especially contracts, focused on delivering a specific end item, often defined by a contract line item number (CLIN), and a corresponding CSCI, CSC, or CSU. An organization, such as test engineering, only considered the particular CSU, CSC, and CSCI being tested. There was little consideration of the value of automatically linking the CLIN to the CSU, CSC, or CSCI to the documentation status and the performance measurements to facilitate management control.

The additional issue identified by the analysis was related to identifying organizational responsibility, e.g., design engineering, by CSU, CSC, or CSCI. The main purpose of performance measurement techniques, such as breadth of test (testing the requirements), is to provide organization members and management the status and potential problems to enable project management to facilitate and assign corrective action to each organization. For example, if testing is behind schedule for a CSU, the project manager must be alerted, who then determines the reason and prepares to assign corrective action to the testers and other relevant organizations.

Essentially, the analyses revealed that the documentation control and performance measurements of the CMM and the PSM were effective except for the following shortcomings: There was no method to identify each type of document and no mechanism to link the CLIN, documentation, performance measurement, and organizational responsibility by CSU, CSC, and CSCI. These shortcomings led to the creation of the DMM.

Design Maturity Model

The DMM resides in EXCEL and provides a framework to prevent loss of control during software development. The DMM enhances the software industry's approach to control software development by using the features of a Product Work Breakdown Structure (PWBS) as a mechanism to integrate and link the documentation, performance measurements, and organizational responsibilities. This analytical approach should enhance organizational effectiveness, facilitate corrective action, and increase the probability of project success. A management control document (MCD) is designed as the primary configuration control mechanism to monitor the documentation status. The initial step is to define the PWBS and link the CSUs, CSCs, and CSCIs to each PWBS element. The next step is to define and link the corresponding documentation. Once the data is entered into the MCD, it can be automatically transmitted to the appropriate performance measurement spreadsheets. In addition, a PWBS/organizational matrix is designed to identify organizational responsibilities for the total system, the CSU, the CSC, and the CSCI. A PWBS/organizational matrix serves as a link to a contractor's accounting system, which is the official financial status of a project and is essential to determine profit and loss.

Project Inception

Documentation control must begin at project inception. During project inception, the customer's and the contractor's most challenging task is to adequately define and document the requirements and expand and flow those requirements into the specifications, functional description, system interfaces, and test requirements. Unfortunately, the software complexity often prevents stabilizing the requirements during this early phase. This instability of the requirements or design forces most contractors to continuously refine the requirements and to build the software incrementally—which is referred to as evolutionary acquisition or incremental builds. As the project evolves, the design matures and documentation control requires measuring the design changes imposed upon the baseline and ensuring that the changes are incorporated or flowed into all the documents. Evolutionary acquisition enforces the idea of the increased discipline inherent in the DMM. The CSUs, CSCs, and CSCIs are linked to the documentation, performance measurements, and organizational responsibility to prevent loss of control as the design matures. The problem of controlling what is being built and linking it to the documentation and performance measurement and who is responsible for building the product surfaced in the 1970s because various contractors failed to fulfill technical and cost objectives. In response to this problem, the Department of Defense (DoD) created a control system: A PWBS defined what was to be built and linked the PWBS to documentation and performance measurements. An organizational matrix linked the PWBS to the organizations or who was responsible for building the product. The PWBS system has proven invaluable as a mechanism to monitor and control technical performance, cost, and schedule.

Product Work Breakdown Structure

Since its creation, the DoD and contractors have successfully used the PWBS system to define and monitor aircraft configurations and to link the corresponding requirements and specifications to the system, subsystem, elements, and organizational responsibilities. Most contractors establish an MCD similar to that shown in Figure 1. It identifies the PWBS for the total system as 1000. PWBS 1000 is level 1 as noted by the arrow in the illustration. The description for level 1 is the total system; the total system name is the Weapon System. The SOW paragraph is 1000. PWBS 1000, level 1 includes level 2: 1100 and 1600. Level 2 includes level 3: PWBS 1100 includes 1110, and PWBS 1600 includes 1610, 1620, 1630, and 1640. Level 3, PWBS 1110, includes level 4: 1111, 1112, 1113, 1114, and 1115. In this example, there is no level 4 for 1600. One point is worth noting: PWBS 1600 is defined as management activities for the entire system and ordinarily is not assigned to any particular PWBS. Each PWBS is assigned a specific SOW paragraph, e.g., PWBS 1111 corresponds to the requirements paragraph 1111. Thus, in clear terms, the system design is defined, and a new PWBS or paragraph may be inserted into the system to accommodate design changes, and it is easy to link the project's documentation and performance measurement techniques to a specific PWBS. However, for software acquisition, the MCD should be more extensive.

Management Control Document Software

The first step toward establishing the DMM's process or model is to assign the CLIN and the PWBS along with its description. The PWBS identifies the configuration or product design by

Figure 1. *Management control document.*

PWBS & LEVEL		Hardware Level	PWBS Description	Requirements Paragraph
1 0 0 0	1	Total System	Weapon System (WS)	1000
1 1 0 0	2	System 'A'	Mission Objectives (MO)	1100
1 1 1 0	3	Subsystem	Engineering Objectives (EO)	1110
1 1 1 1	4	Element 1	System Architecture (SA)	1111
1 1 1 2	4	Element 2	Subsystem Architecture (SSA)	1112
1 1 1 3	4	Element 3	System Interfaces (SI)	1113
1 1 1 4	4	Element 4	Legacy Systems (LS)	1114
1 1 1 5	4	Element 5	Reuse/COTS (RC)	1115
1 6 0 0	2		Management	1600
1 6 1 0	3		Program Management	1610
1 6 2 0	3		Financial Management	1620
1 6 3 0	3		CSCI Manager	1630
1 6 4 0	3		Project Support	1640

software levels: CSU, CSC, or CSCI. The PWBS data is entered into the MCD. The MCD acts as the primary configuration control mechanism. The second step consists of defining the documentation by PWBS and entering that data into the MCD. Documentation control should be concerned with monitoring and controlling the baseline and the effect a change in one document may have upon another: A change in requirements may affect the specifications and the functional description. To enhance organizational communication and mitigate any confusion between documents, a unique letter identifies each type of document: ‘R’ for the requirements, ‘S’ for the specifications, ‘F’ for the functional description, ‘N’ for the system interfaces, and ‘T’ for the test requirements. The MCD data are automatically transmitted or mapped to the performance measurements, and a change entered into the MCD will be automatically reflected on every performance measurement spreadsheet. The MCD enhances ISO 9000 standards. The PWBS enhances management control by identifying the current design and the product(s) to be delivered to the customer, which addresses ISO 9000’s management responsibility. In addition to identifying the current design or product, the MCD provides the means to monitor and control the documentation that reconciles ISO 9000’s product identification and traceability, design control and documentation, and data control. An example follows, which includes solving the problem of evolutionary acquisition or incremental builds.

Figure 2 reflects DMM’s MCD. A CLIN and a PWBS for the software levels CSCI, CSC, and CSU, along with a description, is assigned. Each type of document paragraph is identified with a unique letter, and the paragraph number is identical to the PWBS. The ‘In’ column is used to designate the baseline or incremental build. The baseline is initialized to ‘0’; each increment is indexed. The ‘CH’ column is used to designate design changes imposed upon the baseline (core system) or an incremental build. Figure 2 represents a theoretical weapon system. For this example, an incremental build is assigned to PWBS R1112, level 4. In addition, for PWBS R1111, the ‘CH’ column reflects that there is a change in requirements imposed upon the basic system: The

Figure 2. *Management control document data.*

Management Control Document Data

Contract Line Item Number (CLIN) = 0001

ISO 9000: Management Responsibility, Design Control, Documentation and Data Control and Product Identification and Traceability

Requirements: Reqts
Specifications: Sp
Functional Description: FD
System Interface: SI
Test Plan: TR

Baseline = 0, Design Changes = Number > 0 Increment (In No.) or Build Number = Number > 0

Baseline Increment No. 1 Design Change

PWBS	LEVEL	Software Level S/W Lev	PWBS Descr.	Req'ts Para. 'R'	Baseline or Incr.No 'In No.'	Ch#	Sp Para. 'S'	Ch#	FD Para. 'F'	Ch#	SI Para. 'N'	Ch#	TR Para. 'T'	Ch#
1000	1	Sys	WS	R1000	0	0	S1000	0	F1000	0	N1000	0	T1000	0
1100	2	CSCI	MO	R1100	0	0	S1100	0	F1100	0	N1100	0	T1100	0
1110	3	CSC	EO	R1110	0	0	S1110	0	F1110	0	N1110	0	T1110	0
1111	4	CSU	SA	R1111	0	1	S1111	0	F1111	0	N1111	0	T1111	0
1112	4	CSU	SSA	R1112	1	0	S1112	0	F1112	0	N1112	0	T1112	0
1113	4	CSU	SIA	R1113	0	0	S1113	0	F1113	0	N1113	0	T1113	0
1114	4	CSU	LS	R1114	0	0	S1114	0	F1114	0	N1114	0	T1114	0
1115	4	CSU	RC	R1115	0	0	S1115	0	F1115	0	N1115	0	T1115	0

‘CH’ column is ‘1.’ In other words, PWBS R1111’s requirements have been reviewed, but the specifications, test requirements, etc., have not been reviewed: The ‘CH’ columns remain ‘0.’ Upon reviewing the MCD spreadsheet, any organizational member would be alerted that there was a change in the requirements paragraph R1111, but the change has not been incorporated into the remaining documents. This DMM procedure would ensure an activity such as testing would be implemented based upon the current documentation and configuration. In general terms, the appropriate contractor’s organization—usually the Change Board members—would review, reject, or approve the design changes that affect the basic system or the incremental builds and would be required to update the configuration and documentation status. The same principles used in the MCD, related to documentation control, may be applied to a performance measurement such as breadth of test.

Breadth of Test (BOT)

The performance measurement BOT measures the requirements that have been tested, passed testing, and failed testing. Figure 3 demonstrates how DMM’s BOT would be used in a contractor’s environment. The ISO 9000 quality standards are, essentially, related to testing. In addition, since the documentation status is reflected on the spreadsheet, the BOT spreadsheet reconciles ISO 9000’s Design Control, Documentation & Data Control, Product Identification and Traceability, and Inspection and Testing, which are inherent in monitoring test status. The documentation status, e.g., requirements, are automatically transmitted from the MCD to the BOT measurement spreadsheet. The MCD generates the identical documentation status to all the performance measurements mentioned previously, such as depth of test, computer resources, and complexity. The example includes identifying the CLIN and demonstrates, by PWBS, the status of the documentation to ensure that a tester may verify that testing will be based upon the correct configuration. The ‘In’ column is maintained to identify the baseline, and the ‘CH’ column accommodates design changes. The number of requirements tested, passed testing, and failed testing or defects/faults are included in the example. There are 500 total requirements (PWBS R1000). Of those, 375 have passed testing and 125 failed, or there are 125 defects. The ‘In’ column for R1112 contains a ‘1,’ which indicates an incremental build, and there is a change in requirements for PWBS R1111 (the ‘CH’ column is 1), but the remaining documentation, in particular the test plan for PWBS T1111, reflects ‘0’ in the ‘CH’ column. This implies the responsible organization, usually the change board, has failed to review the test plan for PWBS 1111.

PWBS/Organizational Matrix

The PWBS/organizational matrix permits management to measure organizational performance. The PWBS system was designed to provide a mechanism to roll up the design status and the performance measurement values to each higher level: Level 1 includes level 2, etc. This roll-up feature permits project management the freedom to measure progress at any level: the

Breadth of Test — Data as of MM, DD, YY

Contract Line Item Number (CLIN) = 0001

ISO 9000 Quality System, Design Control, Documentation & Data Control, Inspection & Testing

Requirements: Reqts — System Interface: SI
Specifications: Sp — Test Plan: TR
Functional Description: FD

Td: Tested. P Tg: Passed Testing. F Tg: Failed Testing

Baseline = 0, Design Changes = Number > 0 — Increment (In No.) or Build Number = Number > 0

Baseline — Increment No. 1 — Design Change

PWBS LEVEL	S/W Level	PWBS DS	Reqts Para R'	In #	Ch #	Sp Para S'	Ch #	FD Para F'	Ch #	SI Para N'	Ch #	TR Para T'	Ch #	Breadth of Test Requirements Testing Td	P Tg	F Tg
1 0 0 0 1	Sys	WS	R1000	0	0	S1000	0	F1000	0	N1000	0	T1000	0	500	375	125
1 1 0 0 2	CSCI	MO	R1100	0	0	S1100	0	F1100	0	N1100	0	T1100	0	500	375	125
1 1 1 0 3	CSC	EO	R1110	0	0	S1110	0	F1110	0	N1110	0	T1110	0	500	375	125
1 1 1 1 4	CSU	SA	R1111	0	1	S1111	0	F1111	0	N1111	0	T1111	0	100	75	25
1 1 1 2 4	CSU	SSA	R1112	1	0	S1112	0	F1112	0	N1112	0	T1112	0	100	75	25
1 1 1 3 4	CSU	SIA	R1113	0	0	S1113	0	F1113	0	N1113	0	T1113	0	100	75	25
1 1 1 4 4	CSU	LS	R1114	0	0	S1114	0	F1114	0	N1114	0	T1114	0	100	75	25
1 1 1 5 4	CSU	RC	R1115	0	0	S1115	0	F1115	0	N1115	0	T1115	0	100	75	25

Figure 3. *Breadth of test.*

total system, CSCI, CSC, and CSU. ISO 9000 Management Responsibilities are assigned to PWBS 1600, usually an industry standard for management. Ordinarily, management activities are not allocated to any specific PWBS, but rather to management of the entire system. The PWBS/organizational matrix (Figure 4) represents a theoretical weapon system and defines the PWBS level, CSCI, etc., and assigns an organizational letter, a departmental number, which is the link to the accounting system, and the accounting name of each department. The PWBS/organizational matrix is the essential link from the PWBS to the PWBS descriptions to the organizations responsible for producing the product and to the accounting system. In the example, the organizational responsibilities are reconciled to each PWBS. For example, the program manager, 'a,' is assigned to PWBS 1610, the financial manager, 'b,' to 1620, etc. The PWBS 1600, as well as PWBS 1100, are rolled up to level 1: PWBS 1000.

Essentially, the PWBS/organizational matrix is a valuable tool or report to facilitate management control and define organizational responsibility and authority through all organizational levels. It also serves as the link to audit contract status by organization. Once the metrics measurements and other relevant performance data are identified by PWBS, the project manager can determine what organizations are assigned to each PWBS and use that information to validate responsibilities and assign the required corrective action. In addition, since the PWBS, the level, and the description are automatically generated by the MCD, the PWBS/organizational matrix will reflect the current design status or configuration.

Figure 4. *PWBS/organizational matrix.*

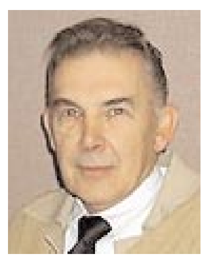
PWBS/Organizational Matrix
ISO 9000 Management Responsibility
(Organizational Responsibility)

PWBS LEVEL	S/W Level	PWBS Description	Organizations a b c d e f g h i j k l m n	Accounting Dept. No.	Department Name	ORG
1 0 0 0 1	System	Weapon System		100	Project Manager	a
1 1 0 0 2	CSCI	Mission Objectives	c d e f h i	110	Financial Manager	b
1 1 1 0 3	CSC	Engineering Objectives	c d e f h	120	Design Engineering	c
1 1 1 1 4	CSU	System Architecture	c d e f h	130	Sys. Engr. & Integration	d
1 1 1 2 4	CSU	Subsystem Architecture	c d e f h	140	Test Engineering	e
1 1 1 3 4	CSU	System Interface Architecture	c d e f h	150	Programming	f
1 1 1 4 4	CSU	Legacy Systems	c d e f h	160	Testers	g
1 1 1 5 4	CSU	Reuse, COTS	c d e f h	170	Database	h
1 6 0 0 2		Management		180	CSCI Manager	I
1 6 1 0 3		Program Management	a	190	Metrics	j
1 6 2 0 3		Financial Management	b	200	Configuration Mgmt	k
1 6 3 0 3		CSCI Manager	I	210	Data Management	l
1 6 4 0 3		Project Support	j k l m n	220	Scheduling	m
				230	Product Assurance	n

Conclusion

Documentation control and performance measurement techniques already are industry standards. Since the DMM will enhance those standards, it should be acceptable to the industry, and there should be little incremental cost to implement the model. DMM simplifies the integration of ISO 9000 standards and improves organizational discipline and documentation control. The DMM may be used by any contractor or customer (DoD or NASA) to enhance and facilitate software development controls. The implementation of the DMM will decrease the cost of software development by reducing the time spent by organizations dedicated to documentation control, reviewing the design status, and ensuring that activities are focused on the correct design and documentation. There are additional savings associated with the increased communication between the contractor and the customer using the MCD, performance measurement data, and the PWBS/organizational matrix. The DMM has been given to the Goddard Space Flight Center through the technology transfer program for possible implementation and to the Langley Research Center and the Kennedy Space Center for review. ◆

About the Author



Donald Przebowski is employed by Decisive Analytics as a senior cost analyst and technical writer and participates in the technical and cost evaluations that pertain to National Missile Defense. He also creates procedures and models that will facilitate the control of software development. Employed by Fairchild Republic for about 22 years, he managed software systems development, parametric estimating, performance measurement, statistics, and major proposals and was a project and business manager on a major subsystem. He joined the Grumman Corporation in 1987 to support the NASA Space Station Freedom Program, where he developed independent cost assessments for hardware and software and evaluated the impact of various design changes. He participated in the evaluation of Boeing, McDonnell Douglas, and Rocketdyne's management policies. He was employed by the Statistic Corporation on a software development project, where he applied performance measurement techniques to measure project status.

Decisive Analytics
1235 Jefferson Davis Highway, Suite 400
Arlington, VA 22202
Voice: 703-414-5032
Fax: 703-414-5066
E-mail: djp@decisive-analytics.com

Privatization of the Web: Hidden Economic Backhand

J. Michael Brower
Department of Justice, Immigration, and Naturalization

If the government forfeits its oversight over the private company currently monopolizing the registration of domain names, the private sector cannot ensure that the World Wide Web remains creative, tax-resistant, and a free-fire zone for new ideas. This article discusses the potential price increases and certain loss of what public control remains over the Web's general direction. Ensuring cheap, public access to the Web implies keeping that resource under the U.S. government and away from privatization.

YOU MAY NOT BE interested in the privatization of the World Wide Web, but Web privatizers are interested in you. Information technology (IT) leaders, such as the GartnerGroup (www.gartner.com), are increasingly interested in apprising Web consumers of the multibillion-dollar potential of the Internet and the Web.

Even utilities, traditionally under the auspices of the public sector, have launched ambitious multimillion-dollar Web-based strategies to capture the imaginations and are clambering into the pocketbooks of online customers. The Herndon, Va.-based utility Columbia Energy, Inc., for example, is riding the energy supplier deregulation wave, cutting snail-mailings and the number of hapless service representatives while enlisting the Net literate (see www.atlantaenergy.com and www.georgiaenergy.com) [1].

John Chambers, chief executive officer of Cisco Systems—one of the fastest-growing Internet developing companies—indicated that by 2010, 25 percent of global commerce will be transacted over the Internet [2]. In a November 1998 report, Forrester Research indicated that Internet-commerce revenues will account for 6 percent of all retail sales in the United States by 2003. Yet today, the Internet and the Web are only babies, developmentally and economically. Also reported in the popular IT press was that 40 million surfing households will spend $108 billion online by 2003, up from $7.8 billion spent by 9 million households in 1998 [3].

Enter the Web privatizer.

To them, the numbers indicate that Internet economics will drive Web developments, such as languages and applets, that are pursued rather than the arguably pro-public developments, e.g., education and freeware. The systemic stripping of national stewardship over the American-sponsored Internet is frittering away our most significant, taxpayer-underwritten, communication accomplishment of the 20th century.

Ironically, the Clinton administration's ongoing Internet commerce initiative would include the establishment of federal regulation to protect online buyers [4]. Despite this defense of the online public, privatization culminates in the very monopolistic business configurations that increase costs to Web frequenters, public and private. Web taxation and government over-regulation are anathema to all progressives, but unbridled privatization will drive up costs and in so doing, make access more exclusive. The culmination of the Web privatizers' handiwork can be glimpsed through the publication of the first annual report (November 1998) of the U.S. Government Working Group on Electronic Commerce (http://www.doc.gov/ecommerce/review.htm) and the announced departure of Ira C. Magaziner, adviser to President Clinton on Internet affairs. Abandoning public oversight of the Internet receives an overhasty nod in *Electronic Commerce* report, and Magaziner's departure[1] is a clear declaration of private-sector victory over the public's interest in the Web. Internet czar Magaziner "successfully" arranged the current struggle by various companies for the mantel of distributor and ultimate controller of domains .com, .net, .org, and others. This is despite assurances that a nonprofit organization is sought for this important mission [5]. Notwithstanding, the furious maneuvering so rudely following the untimely death of Internet godfather Jonathan B. Postel could only be so impassioned over one thing—money. With Internet use doubling every 100 days and an estimated 100 million worldwide users online regularly, the rules that Postel proposed—hostile to for-profit privatization of the Web—will hardly survive him against conspicuous commercialization.

Through the Commerce Department's National Telecommunications and Information Administration (NTIA), the Clinton administration proposed a nonprofit corporation to manage domains in a June 1997 policy paper published by NTIA.[2] The new Internet Corporation for Assigned Names and Numbers (ICANN), eclipsing the Internet Assigned Numbers Authority (IANA)—a government contractor in Marina Del Rey, Calif.—will control work formerly done exclusively via government contract by Network Solutions of Herndon, Va.

For the moment, IANA will continue to issue numerical IP addresses, and Network Solutions will administer domain name services. Network Solutions is loath to surrender its generally benign, oligopolistic partnership with the federal government.[3] The consolation prize for Network Solutions is the continued control over the domains it has distributed. With privatization,

An edited and shortened version of this article appeared in Computerworld, *March 8, 1999, pg. 36, http://www.computerworld.com/home/print.nsf/all/9903089432*

it will perhaps be capable of exerting financial leverage it could not exert given public oversight. Network Solutions will retain its monopoly over the domain names it has already given out, but competitions must be held for new business in this area.[4] But just as private enterprises can be bought or sold, so too, it seems, can the *Made in the U.S.A.* label the Internet tenuously retains. Will the new arrangement guarantee that the Internet's future will not be dictated via company acquisition or hostile takeover by a foreign company—British, German, or Japanese? It is a genuine prospect under the current privatization formula.

The domain name-controlling organization as currently configured is considered by many to be insufficiently open and anti-democratic. One commentator on the Web's privatization, Ronda Hauben, put it aptly when quoted in a *Government Computer News* article [6], "Privatization would be moving policy functions out of the control of government and putting them into unaccountable hands. The whole result of this is very dangerous for the public and the Internet."

Privatization vs. Piratization

Privatization is not new—Adam Smith was writing about it in 1762. The British South Africa Company and the Dutch East Indies Company were in private hands until they were absorbed to support global imperialism in the 19th century [7]. Small wonder that important public functions move from government to private control, and back again, with changing times. But Web privatization is akin to the malfeasant genre of gangster capitalism ravaging so much of the former Soviet bloc. The fetish of turning over the publicly underwritten to private hands is based on the industry-manufactured perception that state control seldom achieves public benefits at the lowest possible cost. Nothing of the kind. The moral nomads never mention privatization's many disasters and false starts [8]. Suspending the U.S. public sector's Internet oversight role ignores, for instance, the great strides that public sector chief information officers have taken in recent years and their ability to infuse the public interest into the for-profit milieu of the Web [9].

The contract the government has with Network Solutions has been extended until autumn of 2000. The privatization effort was to be consummated by Sept. 30, 1998 but—fortunately—is still under study [10]. For the moment, IANA will continue to issue numerical IP addresses, and Network Solutions, Inc. will administer domain name services. It is not too late—nor is it mere neo-pax Americana—to suggest that the Internet be declared a strategic resource by its creator, the U.S. government, and not be left to possible domination by a foreign entity through market manipulation. No nation should "control" the Internet—but neither do Internet economics allow the United States to afford a global, private sector dictatorship of this indispensable public resource. Maintaining the root server system that maps the domains to IP addresses must stay within the grasp of the same American public whose taxes originally underwrote the Internet. This piece of IT is ours, and it should not be for sale. ◆

About the Author

J. Michael Brower is a Vermont writer. His previous assignment was at the Pentagon with the Office of the Assistant Secretary of the Army (Financial Management and Comptroller), Army Business Practices Directorate. He is currently at the Department of Justice, Immigration, and Naturalization Service.

E-mail: jmichael@together.net
Internet: http://www.geocities.com/capitolhill/lobby/2985/

References

1. Machlis, Sharon, "Utility Powers Up Net Strategy to Lure Customers," *Computerworld*, Dec. 7, 1998, p. 4.
2. Weil, Nancy, "Clinton Spells Out I-Commerce Plan," *InfoWorld*, Dec. 7, 1998, p. 67.
3. "Focus on I-Commerce," *InfoWorld*, December 7, 1998, p. 79.
4. ibid., p. 67.
5. Harmon, Amy, "U.S. Expected to Support Shift in Administration of the Internet," *The New York Times*, Oct. 20, 1998, p. C1.
6. Jackson, William, "Commerce Settles on a Plan for Net's Privatization," *Government Computer News*, p. 56, Oct. 26, 1998.
7. Cowan, Laing Gray, *Privatization in the Developing World*, Praeger Publishers, 1990, p. 7.
8. For some of privatization missteps, see "Privatization and the Defense Worker's Opposition," Program Manager, Sept.-Oct. 1996, p. 19; Defense Systems Management College, http://www.dsmc.dsm.mil/pubs/pdf/pmtoc96.htm
9. Fabris, Peter, "Odd Ducks No More: Public Sector CIOs Are Now Key Players in the Reinvention of Government," CIO, Nov. 15, 1998, p. 41. http://www.cio.com
10. Gerwig, Kate, *InternetWeek Newsletter* of Feb. 2, 1998 via an Army listserve, 53list serv@orator.usma.edu

Notes

1. Magaziner will probably be succeeded by David Beier, Vice President Gore's chief domestic policy adviser (source: *Associated Press newswires*).
2. This white paper, the "Framework for Global Electronic Commerce" released in July 1997, urged governments not to create taxes for at least three years. The privatization effort was expected to be sustained. (*InfoWorld*, Nov. 16, 1998, p. 62, story by Bob Trott, "Presidential Internet adviser leaving post").
3. Network Solutions generated revenue of $37 million in the first half of this year by registering names with the .com, .net, .org, and .edu suffixes—not an easy take to part with without a struggle.
4. Ideally, the "registrars" or the private companies that would register domain names would all report to another not-for-profit entity overseeing a reorganized IANA that maintains the master database of numerical Internet addresses that support Web addresses.

The Acid Test: Measuring Your Success

How do you measure success? How do you know if you are a successful software engineer? What's the yardstick—cost, schedule, defects, time out of meetings, pizzas devoured per project? What is the ultimate measure for defense software engineers?

In other professions, it seems the ultimate measure is more definitive. For Lucas and Spielberg, it is ticket revenue, despite the spiel you hear about artistic satisfaction and golden statues. For O'Neil and Malone, it is a ring like Mike's. For Woods, Duvall, and Olazábal, it is the green jacket that only a valet would wear. For Armstrong and the cast of thousands at NASA, it was that "small step for mankind."

It is interesting that those who excel in their profession have one ultimate goal or measure that directs, motivates, and defines their success. A measure that eclipses all others. It is often referred to as the show, the dance, the big enchilada, the bottom line, or as my chemistry teacher put it, "the acid test."

Measures necessary in preparation, guiding, and managing a project are fruitless if you fail the acid test. Duke's individual and team statistics, although superior, have no luster compared to Connecticut's Championship Trophy. If Star Wars movies languished at the box office in the '70s, there are no sequels, prequels, or Lucas Films, and we would not be experiencing the magic of "The Phantom Menace" in theaters today. Likewise, in defense software engineering, cost, schedule, defects, and that Holy Grail called capability maturity pale in importance to customer satisfaction.

Who are our customers? What satisfies them? What is our acid test? We are in the defense industry. Our customers are warriors. Their satisfaction is the ability to accomplish their mission. Their current mission, our acid test, is occurring in the Balkans and reported on the front pages of many publications in America. How are you doing?

Those who supply A-10 pilots with Global Positioning System (GPS) receivers received their report card from the Balkans. It appears their customers decided to use their own funds to purchase off-the-shelf GPS receivers. They included them in their survival gear and attached them to their cockpits with Velcro. Why? Because the government-furnished Combat Survivor Evader Locator (CSEL) radio, hailed by the Department of Defense as a "success story," is not available. Nice acronym, acid test failed.

What is more disturbing are the excuses offered in the March 29 issue of *Federal Computer Week*. CSEL defenders warned against the use of commercial GPS equipment because the systems may not be reliable, are susceptible to jamming, and have no protection against spoofing. True, but at least it is in the dance! Pilots have spoken; they would rather go to war with a GPS receiver susceptible to jamming than a nonexistent CSEL.

This happens with software, too. When I was designing operational flight programs for the F-16 Fighting Falcon, mission-planning systems were being introduced to reduce time and errors in flight planning. The program office's solution for these mobile planning systems resembled Molly Brown's steamer trunk with the mobility of a beached whale. A pilot from the Air Force Reserve took a laptop computer, commercial database, and a C compiler and prototyped a system that ran circles around the current system in mobility, functionality, and usability. The program office lodged complaints about susceptibility, survivability, and reliability. Regardless, the pilot shed light on a design that reached beyond the drawing board and into the reality of war. He understood the acid test.

How is your report card from the Balkans? Is your customer satisfied? How do you know? Are you collecting customer feedback from the Balkans? It is a good thing you cannot Velcro software to a cockpit.

—Gary Petersen, TRI-COR Industries

Got an idea for BACKTALK? Send an E-mail to backtalk@stsc1.hill.af.mil

CROSSTALK

SPONSOR *Lt. Col. Joe Jarzombek*

PUBLISHER *Reuel S. Alder*

ASSOCIATE PUBLISHER *Tracy Stauder*

EDITOR IN CHIEF *Brent Baxter*

SENIOR EDITOR *Sandi Gaskin*

MANAGING EDITOR *Kathy Gurchiek*

ASSOCIATE EDITOR/LAYOUT *Heather King*

ASSOCIATE EDITOR/FEATURES *Heather Winward*

VOICE 801-775-5555 DSN 775-5555
FAX 801-777-8069 DSN 777-8069
E-MAIL crosstalk@stsc1.hill.af.mil
STSC ONLINE http://www.stsc.hill.af.mil
CROSSTALK ONLINE http://www.stsc.hill.af.mil/Crosstalk/crostalk.html
ESIP ONLINE http://www.esip.hill.af.mil

Subscriptions: Send correspondence concerning subscriptions and changes of address to the following address:
Ogden ALC/TISE
7278 Fourth Street
Hill AFB, UT 84056-5205
E-mail: custserv@software.hill.af.mil
Voice: 801-775-5555
Fax: 801-777-8069 DSN 777-8069

Editorial Matters: Correspondence concerning Letters to the Editor or other editorial matters should be sent to the same address listed above to the attention of *CROSSTALK Editor* or send directly to the senior editor via the E-mail address also listed above.

Article Submissions: We welcome articles of interest to the defense software community. Articles must be approved by the CROSSTALK editorial board prior to publication. Please follow the *Guidelines for CROSSTALK Authors*, available upon request. We do not pay for submissions. Articles published in CROSSTALK remain the property of the authors and may be submitted to other publications.

Reprints and Permissions: Requests for reprints must be requested from the author or the copyright holder. Please coordinate your request with CROSSTALK.

Trademarks and Endorsements: This DoD journal is an authorized publication for members of the Department of Defense. Contents of CROSSTALK are not necessarily the official views of, or endorsed by, the U.S. Government, the Department of Defense, or the Software Technology Support Center. All product names referenced in this issue are trademarks of their companies.

Coming Events: We often list conferences, seminars, symposiums, etc., that are of interest to our readers. There is no fee for this service, but we must receive the information at least 90 days before registration. Send an announcement to the CROSSTALK Editorial Department.

STSC Online Services: STSC Online Services can be reached on the Internet. World Wide Web access is at http://www.stsc.hill.af.mil.
Call 801-777-7026 or DSN 777-7026 for assistance, or E-mail to schreifr@software.hill.af.mil.

Back Issues Available: The STSC sometimes has extra copies of back issues of CROSSTALK available free of charge. If you would like a copy of the printed edition of this or another issue of CROSSTALK, or would like to subscribe, please contact the customer service address listed above.

The Software Technology Support Center was established at Ogden Air Logistics Center (AFMC) by Headquarters U.S. Air Force to help Air Force software organizations identify, evaluate, and adopt technologies that will improve the quality of their software products, their efficiency in producing them, and their ability to accurately predict the cost and schedule of their delivery. CROSSTALK is assembled, printed, and distributed by the Defense Automated Printing Service, Hill AFB, UT 84056. CROSSTALK is distributed without charge to individuals actively involved in the defense software development process.

Measures and Metrics

An Effective Metrics Process Model — *Capt. Thomas Augustine, Charles Schroeder* 4
Poorly contrived measurements can make for poor decisions.

PSM Insight: The Army-DoD Tool to Implement Issue-Driven Software Measurement — *Don Scott Lucero* 8
A tool with the flexibility to support an effective, tailored measurement process.

It Is the People Who Count in Measurement: The Truth about Measurement Myths — *Carol Dekkers, Mary Bradley* 12
How people issues can cripple a measurement program or lead to its resounding success.

CMM Achievements

USDA's National Finance Center: A 10-Month Journey to Level 2 — *A. Todd Steadman* 15
Implementing the necessary software process improvements required to achieve Level 2.

Software Engineering Technology

Mission-Critical and Mission-Support Software: A Preliminary Maintenance Characterization — *Elizabeth K. Bailey Clark, James A. Forbes, Emanuel R. Baker, Donald W. Hutcheson* 17
An initial characterization of Department of Defense mission-critical software maintenance.

PAIR: A Rational Approach to Fighting Software Project Fires — *Gregory T. Daich* 23
Putting out critical defects fires before they start.

Emerging Ideas

Design Maturity Model — *Donald Przebowski* 25
Enhance documentation control and performance measurement techniques.

Open Forum

Privatization of the Web: Hidden Economic Backhand — *J. Michael Brower* 29
Frittering away the most significant communication accomplishment of this century.

Departments and Announcements

2 **From the Publisher:** The Need for a Measurement and Analysis Process: Focusing on Guidance for Process Improvement
7 **Measures and Metrics Web Sites**
16 **FMSO Achieves CMM Level 4**
31 **BackTALK:** The Acid Test: Measuring Your Success

Letter to the Editor 3
Coming Events 3
Call for Articles 14

CROSSTALK
Ogden ALC/TISE
7278 Fourth Street
Hill AFB, UT 84056-5205

BULK RATE
US POSTAGE PAID
Permit No. 481
Cedarburg, WI