From the Publisher

Sharpening Your Management Skills

Tracy Stauder
Software Technology Support Center

I recently attended a project management workshop. Prior to attending this workshop, I would not have considered myself a manager. But as I listened to several days of lectures, it became clear that we are all managers. We manage budgets, contracts, projects, programs, processes, software, time, risk, stress, change, money, our children, and ourselves.

Whether at home or work, whether we manage people or things, there are some fundamental management techniques to keep in mind. Communicating, listening, motivating, planning, organizing, evaluating, coaching, problem solving, and decision making are a few of the skills we need in our tool kits [1].

Also, we tend to learn skills from observing others. What traits does your current or former manager have that you admire or dislike? Perhaps fairness, honesty, credibility, timeliness, trustworthiness, and professionalism might be on your list of desirable traits. What do you expect from your manager? Chances are these are the same things that your employees or co-workers expect from you.

Take a minute and think about which of these basic management skills you effectively utilize on a day-to-day basis and which you struggle to use. Pick a few techniques you can practice and sharpen. By focusing on a few each day or week, you can increase your management performance and professional growth.

In this month's issue, Quentin Fleming and Joel Koppelman (see page 10) present a project management storybook tale. Although a tale, it presents real-life management challenges and shows the importance of basic management techniques, such as planning and earned value. As the article shows, earned value performance measurement is a tool to help managers evaluate schedule and budget, and it can be especially useful in the early phases (15-20 percent project completion) of a project's life cycle.

Standards and models also are good methods to help us improve our management skills. The article by Jeremiah Smedra (see page 15) will introduce you to the Project Management Body of Knowledge (PMBOK) model. This model is recognized widely as the commercially proven and accepted standard for project management. The PMBOK model gives a manager a proven framework for project success.

As managers, we need to find the time to improve our skill set. The better our basic management skills, the better managers we will be. We can get the results needed for our projects and organizations to succeed. I wish you luck in developing and sharpening your management skills. ◆

Reference

1. Johnson, J., "Basic Management Techniques Workshop," April 23, 1999, Technical Management Services™.

Letter to the Editor

Software Development Outsourcing Research Survey

I'm an active duty officer and full-time software engineering doctoral student at Arizona State University. With your assistance, I hope to identify software outsourcing strategies, motivations, benefits, drawbacks, and factors that influence the success of outsourcing relationships. This information will help us understand why outsourcing efforts succeed or often fail to meet goals, and which of these strategies are most appropriate for specific projects and goals. Using this knowledge, I will produce a decision support tool to aid software development project managers and consultants in making software outsourcing strategy decisions for specific projects.

You can help by answering a brief survey if you have participated in a software development project where any portion of the product development or effort has been contracted to an outside vendor. All participants will receive survey results and free copies of the decision support tool. The survey is available at my Web site and can be completed online, electronically using MS-Word, or in paper format.

Maj. Brian G. Hermann, USAF
Arizona State University, Tempe, Ariz.
brian.hermann@asu.edu
software outsourcing Web site: http://www.eas.asu.edu/~outsrc/

Up Close with Dr. Delores Etter

Kathy Gurchiek
CROSSTALK Managing Editor

As Deputy Under Secretary of Defense (Science and Technology) since June 1998, Dr. Delores Etter is responsible for Defense S&T strategic planning, budget allocation, program execution, and evaluation. She also is responsible for the Department of Defense (DoD) High Performance Computing Modernization Organization and the Defense Modeling and Simulation Program.

Her background in the defense community includes chairing the Naval Research Advisory Committee from 1995-97, recent membership on the Defense Science Board, and sitting on the Ballistic Missile Defense Advisory Committee.

She coordinates the North Atlantic Treaty Organization's (NATO) collaborative efforts for science and technology, and is the principal U.S. representative to NATO's Research and Technology Board and to the Technical Cooperation Program among Australia, Canada, New Zealand, the United Kingdom, and the United States.

Etter has long been a part of the education community. She was a professor of electrical and computer engineering at the University of Colorado, Boulder, from 1990-98, and on the faculty at the University of New Mexico from 1979-89. There she was Associate Chair of the Department of Electrical and Computer Engineering from 1987-89. Etter was a National Science Foundation visiting professor in Stanford University's electrical engineering department from 1983-84. She also has spent time at Sandia National Laboratories working in seismic signal processing.

Her educational and research interests include software engineering technologies, development of collaborative experiments in virtual teaming of students, using the Internet, developing distance learning courses for computer software tools, and digital signal processing.

In the next two years, she will also be the executive agent for acquisition of software and has management oversight of the Software Engineering Institute. She is a Fellow of the Institute of Electrical and Electronic Engineers (IEEE), where she has held various positions, including president of the IEEE Acoustic, Speech, and Signal Processing.

Etter is the newest member of the NCS (NASDAQ:NLCS) board of directors. NCS is a global information services company headquartered in Minneapolis, Minn., with employees in North America, Europe, and Australia. It provides software, services, and systems for the collection, management, and interpretation of data. Its focus is serving the kindergarten through high school market.

She can be reached at the Office of the Director, Defense Research and Engineering, the Pentagon, Washington, D.C. Contact also can be made at 703-695-0598 or at etterdm@acq.osd.mil

Editor's note: Dr. Etter was one of the featured speakers during the general session of the 11th annual Software Technology Conference held May 2-6 in Salt Lake City, Utah. *CROSSTALK* had an opportunity to talk with Dr. Etter at length, following her remarks. Here is the result of that interview, interspersed with excerpts from her speech, which are set apart in boxes.

CROSSTALK: During the past several years, the DoD and other government agencies have had no specific objectives or direction dealing with the improvement of software development and sustainment processes and practices. Do you see a more directed and specific direction coming from senior Air Force leadership?

ETTER ON THE IMPORTANCE OF SOFTWARE . . .

"It is true that software is the new physical infrastructure of the information age … Software is everywhere we look within the Department of Defense."

ETTER: It seems to me it's very important that we have more things happening in this area, and that's certainly one of the things I would like to participate in. … But I see it really as something that is going to (come) from working together with all of the services, not just the Air Force, because clearly every one of the services has very major software programs. ... If we are going to come up with some specific goals, (then) they really need to be goals that all of the services are a part of.

CROSSTALK: Former Deputy Assistant Secretary of the Air Force, Lloyd K. Moseman II (now corporate vice president at SAIC) was working with the Air Force and saw strong interest in software and talked about some specific direction. We wondered if you saw that in the DoD and if there was going to be a strong emphasis on software.

ETTER: I think everybody is concerned about it.

There are some retired officers who really have some incredible insight into systems, and one of the first things I would like to do is get together a small group (of flag-level retired officers from each of the services). I plan to go to the service acquisition executives and ask them to give me suggestions of candidates. …(The different organizations within the services and agencies that are working on software-related projects) would meet and address some of these kinds of questions, because I think they would have a lot of good ideas. … The way you get people to buy into that is that they have to see they are a partner in what is going on. It's very critical to have people involved from the services and agencies in both of these groups.

ETTER ON REQUIREMENTS FOR SOFTWARE DEVELOPMENT . . .

"There are a lot of ways we are able to work with discipline, and certainly one of those is through the Capability Maturity Models for software. … Discipline is important on both sides of the street, so I'm looking forward to starting discussions on how we include acquisition within CMMI."

CROSSTALK: Some kind of a steering group from DoD made up of all the Services?

ETTER: I see a very informal advisory group. The other place I think it's important to have a lot of interaction is with all the various software development organizations within the DoD that have a formal organization that provides some kind of software support — either within their service or organization or even on a broad level. I think we are going to be surprised at the number of groups that perform this function.

CROSSTALK: What priorities should software developers, sustainers, and acquirers place on improving their practices and process?

ETTER: It's top priority, not just for the software developers but for the DoD as a whole. It has to do with the discipline. And, of course, discipline isn't just on the software side. There are a lot of pieces to the overall system, but certainly the software is one part of it.

We have to have a lot of discipline and some way of certifying or measuring the changes in the process. I think that's absolutely critical.

CROSSTALK: Are we going to continue with process improvement for software development and acquisition organizations?

ETTER: It really needs to be both sides of the street, doesn't it? If the acquisition side doesn't have the discipline in it, then having a very high level of discipline on the software development side still doesn't get you systems that are on time and on budget and have the functionality that the acquisition side expected, because there's a difference in expectations.

Capability Maturity Model and CMM are registered trademarks of Carnegie Mellon University.

CROSSTALK: One of the conference speakers at the STC '99 general session, Dr. John Guttag of MIT, talked about assembling DoD software instead of coding software. What is your response to that?

ETTER: Well, it's a very interesting concept, isn't it? My interpretation was that it was taking reusability to another level.

CROSSTALK: Is that doable?

ETTER: Sure. That doesn't mean we could do it today, but it's certainly the kind of thing that if we were to make that a goal and begin working towards that, I think it is.

It would take a commitment that that's one of the objectives we want to achieve within the DoD's software program. And it would take commitments of dollars and you are talking about something that's probably in the realm of the S&T program. We certainly are not there today.

CROSSTALK: What kind of timeframe would that involve?

ETTER: I would think a program that was a three- to five-year timeframe would be one that would give you very good insight into what was feasible and also give you some very workable systems.

CROSSTALK: Software process improvement initiatives or mandates within the DoD have become less clear during the past few years. What do you see coming as requirements for software development and sustainment organizations — either government or contractor?

ETTER: I think we have to have some kind of a policy to have consistency. Consistency is very important when you are trying to develop systems that you want to work together and you want reusability. You have to have some consistent policy. But my sense is that the less policy you have, the better. We ought to be able to do things simpler rather than more complicated. Now I know that is an easy thing to say and not necessarily an easy thing to do.

Certainly, that's one of my goals: To see if we can't simplify the policy without giving up discipline … the discipline is absolutely critical.

CROSSTALK: Government organizations don't respond to normal business processes because they don't have a profit motive. USWest, for example, is putting in millions of dollars over the next few years to get to higher levels of maturity because it's going to save them money and they know that. It's quite an investment. But government agencies don't seem to be motivat-

ed by those kinds of things.

ETTER: But it's very important for us to change that sense.

There's a very strong sense within OSD (Office of the Secretary of Defense) right now to look at commercial practices and pick up there what we can and to try to do things that, in the long term, are going to be better.

Not looking at the short term but looking at the longer term. We have to do it in a way that the different groups that are going to be affected are a part of the discussion and whatever policy comes out of it.

What doesn't work is for OSD to mandate a lot of things. What works is to get people together.

CROSSTALK: You are going for a consensus approach?

ETTER: As much as you can. You want people to buy into the process that is going to improve the system, but consensus to some extent really slows things down, so I think you have to find a happy medium. … You have to make things happen.

If all you do is have committee meetings and talk about it, nothing happens. Lots of good ideas get discussed, but nothing gets implemented. On one hand you have to do the consensus building and the partnering, but then you also have to make some decisions and give things a chance to work.

CROSSTALK: Do you see specific initiatives or programs dealing with software and related issues in the future?

ETTER: We shouldn't have such strong separation between the acquisition side and the S&T side. …We have some new things going on in the S&T arena that I think are going to be very important to the acquisition side.

A lot of it is through the programs that are going to be started in fiscal year 2000, part of the information technology initiative.

- There's $70 million of new programs in the Defense Advanced Research Projects Agency. It's more advanced, or applied types, of activity.
- There's $10 million in this multidisciplinary research program. The $10 million is basic research (going into) university/industry collaborations.

We have a lot of things that are going to be starting that cover the whole range of the S&T spectrum. I think we are going to see a number of things that could very possibly lead to changes on the acquisition side.

CROSSTALK: And these are all things we will see in the next fiscal year?

ETTER: They will be starting in 2000. That's not to say thatthere are not things going on right now. It's very clear we need to be doing more in this arena. That's why you are seeing this initiative within DoD to do a very significant increase in the funding we have there.

CROSSTALK: Do you have any thoughts in using distance leaning technology in the DoD? Are you going to push that?

ETTER: It's certainly something the DoD has been using a lot.But I think there are additional ways the DoD can capitalize on things that are coming out of technology.

We have a new initiative in the S&T program that we are calling cognitive readiness. It covers a lot of different areas, for example, the augmented reality. (Dr. Guttag) also mentioned more about it. It covers learner-centric education, which probably is the closest thing to distance learning.

We all learn in different ways. … I'm a visual person; if I can see a diagram or picture of it, I learn it much faster than somebody describing it, which is probably why I'm an engineer.

It's very much in the realm of things today that we can be designing education systems that are computerized — you sit down and you start interacting with it. And by asking questions in certain ways and getting your responses, the software can figure out how you learn best and adapt its whole interaction with you based on that. That's one very simple example of what we are talking about in terms of cognitive readiness.

CROSSTALK: How would you relate that to the DoD in terms of training?

ETTER: We talk a lot about physical readiness, but the education and training side is also a very important part of readiness. (Education and training) also looks at how we augment or add

ETTER ON SPECIFIC PROGRAMS DEALING WITH SOFTWARE FOR FY 2000. . .

" • Multidisciplinary university research program. Some of the areas include real-time, fault-tolerant network-centric protocols. The objective is to develop the foundations of adaptive mobile network protocols for network-centric systems. Requirements include quality of service guarantees of real-time and fault-tolerance performance, security, and safety.

• Interoperability and emergent behavior. We need to develop links that are appropriate for systems with characteristics of adaptability, self-assembly, rapid reconfiguration, self-stabilizing, and fault tolerance.

• Mobile augmented reality. (This) is going to look at recent advances in information technology for mobile use and in novel interactions.

to a person's sense of what the environment is around them. There may be things that real-time sensors are providing; if you can give that to the individual soldier as things are happening, you make a huge difference in the success of missions.

CROSSTALK: A question certainly related to that is what role should the DoD entities play in furthering the sciences and practices related to software?

ETTER: I think it should play a very important role. I think the Software Engineering Institute is a very valuable resource for trying to collect things like that and to build programs that help us be aware of those types of things — run the studies, the analyses. I think the Software Engineering Institute might be a key player in that.

ETTER ON THE TRAINING OF WARFIGHTERS . . .

"You can envision a helmet where as a soldier goes into a building, immediately on the helmet is displayed the layout of that building and shows him exactly where the other people are in the building. This would be done in real-time, so as people move they would (be shown) on the helmet display. It's this kind of information (in real-time) that you are able to give to a person that they wouldn't have otherwise."

CROSSTALK: The DoD started the computer industry. This industry has taken off and the DoD's participation is maybe 10 percent at most. How does the DoD keep up? How does it get what it needs from the industry?

ETTER: Or take advantage of what's out there. It's a real challenge. Even though in many ways DoD is not a key player in driving what happens, you look at the software systems being developed and it clearly has the largest systems and the most expensive systems that are being developed. Somehow it's a key player in that, but we are not driving it and in many ways maybe it's driving us. It has changed a great deal.

CROSSTALK: How are you going to communicate this vision and these kinds of goals to the implementers, the buyers, the developers, the people who are sustaining weapons software, so that they can move toward goal attainment?

ETTER: Program managers are clearly very key people in this. But if you want them to change, to be adaptable and flexible to doing things in a way that you are going to see benefits longer term rather than short term, … you don't just convince them, you also have to have the service acquisition executives buy into it and recognize that it means you have to maybe evaluate these people differently, you have to evaluate the performance of the program differently.

One of the things that I'm also planning to do is ask that we add software reviews to each of our major reviews on acquisition programs. I'm amazed that's not being done now.

CROSSTALK: It's one of the most costly elements (of a program).

ETTER: Absolutely. It comes up when there is a problem, but we ought to be asking questions at every single Defense Acquisition Board review, every single milestone review. That's something that I'm planning.

CROSSTALK: In your position as Under Secretary, if you had to choose one thing, what is your priority?

ETTER: That's a very tough one. This is such a broad program. I think it's really critical that we focus on the things that are going to give the warfighters the revolutionary edge. And that certainly means we have to carefully assess and plan our S&T programs.

One could say that a dramatic improvement in software development capabilities gives one a real revolutionary edge, too. Because if we can't get the systems such that they are affordable and on schedule, some of those revolutionary capabilities that would make a dramatic difference three years from now won't get out there until maybe eight or 10 years from now.

Certainly you could say software performance falls into those categories. That's probably the challenge — to decide what things fall into that category and really focus on them. ◆

OO-ALC/TIS Receives 1998 Gen. Bernard P. Randolph Engineering Team Award

The Software Process Improvement (SPI) team, of the Ogden Air Logistics Center, Software Engineering Division (OO-ALC/TIS), is this year's winner of the 1998 Gen. Bernard P. Randolph Engineering Team Award.

The team was singled out for its exemplary contributions and dedication to the engineering and technical management community, according to James F. Bair, director of engineering and technical management. The award recognizes a team's leadership skills, innovative engineering and process management improvement, customer focus and satisfaction, and engineering/technical merit.

In July, the team achieved a Level 5 Capability Maturity Model rating, unprecedented in the history of the federal government. This accomplishment, which placed OO-ALC/TIS among the top five software organizations in the world, was a major factor to garnering the Randolph Award, according to Lt. Col. Joe Jarzombek.

The TIS SPI team, led by Daniel J. Wynn, TIS Division Chief, is made up of electronic engineer supervisors David Putman (TIS-3), David Haakenson (TISFB), and Bruce Rudd (TISFD); electronic engineers David Webb, Kevin Tjoland, and Mark Peterson (TISF); Bryce Griffin (TISM); Thomas Gompert (TISA); Patrick Cosgriff, Walter Donohoo, and Rusby Craig (TIS-3); Clint Lewis and Lynn Silver (TISEB); Steven Tiede and Kirk Douglas (TISMB); Charles McPhee (TISAD); Steven Philpot (TISAB); Jeff Styers (TISAC), and Capt. Paul Siebels (TISMD). Col. Herbert B. Sherbinske, Hill AFB, nominated the team for the award.

Other awards presented at the banquet hosted at Wright-Patterson Air Force Base in Ohio, were:
Junior Engineer Achievement, won by Capt. Stephen W. Hill, chief, Systems Engineering for ESC/GATL; *Senior Engineer Achievement*, Col. Daniel J. Pierre, director of operations for AEDC/DO; *Outstanding Chief Engineer*, Col. William McCasland, chief engineer for NAVSTAR GPS Joint Program Office for SMC/CZ; *Outstanding Technical Management (individual)*, Capt. Brett G. Scott, deputy program manager, advanced programs, for SMC/ADC; *Outstanding Technical Management (team)*, Global Air Traffic Operations IPT, team chief Maj. Grant Carlson for ESC/GAT; *Outstanding Engineering Technician*, Charles H. McClenahan, Mechanical Engineering Technician for AFRL/MNMF; *Outstanding Production, Manufacturing, or Quality Assurance*, Hamid R. Akhbari, manufacturing systems engineer for ASC/YCDB; *Career Achievement*, Gary L. Bailey, radar systems engineer for ASC/FBJ; *Gen. James Ferguson Award*, Maj. Ronald D. Hackett, electrical engineer for AFRL/DEH; *Capt. Roland R. Obenland Memorial Award*, Capt. Andrew Roberts, lead field support engineer for ASC/LPP.

Gen. George T. Babbitt, Commander, Air Force Materiel Command, presented the awards.

TIS Division Chief Daniel J. Wynn, center, receives the Gen. Randolph Engineering Team award from Gen. George T. Babbitt. James F. Bair, directer of engineering and technical management, HQ AFMC, hosted the banquet and presentation ceremony. Wynn received the award on behalf of the OO-ALC/TIS Software Process Improvement team, which he oversees. The award was presented during the annual Engineering and Technical Management banquet at Wright-Patterson Air Force Base, Ohio. In addition to the plaque, the team's name will be added to the traveling trophy, above left, which the team will keep for one year at Hill Air Force Base, Utah. Photographs taken by Mark Pugel.

Software Technology Conference '99

Kathy Gurchiek
CROSSTALK Managing Editor

The 11th annual Software Technology Conference in Salt Lake City, Utah brought together representatives from industry, government, and academia to increase awareness of software engineering issues and technologies through tutorials, presentations, general sessions, and ad hoc discussions. Interoperability, greater use of software engineering, more emphasis on architecture, reuse of designs and code, and other methods and improvements were among the conference topics. Next year's conference is planned for April 30-May 5. For more information, contact Dana Dovenbarger at dovenbad@software.hill.af.mil

BELOW: David Fugan of Northrop Grumman for Defense Contract Management Command, aims for a hole-in-one at the booth sponsored by Rose International.

LEFT: An STC '99 attendee takes a power nap in the exhibit hall.

RIGHT: Thomas Brandt of the Software Engineering Institute digs into the goodies served up at the informal social in the Salt Palace Convention Center Exhibit Hall.

BELOW RIGHT: Laurie Whitney of Relaxation Station administers a massage at the OC Systems booth.

LEFT: Peggy Ingerski, left, APM; Karen D. Prenger, deputy for operations; and Dr. LorRaine Tauchi Duffy, all of SPAWAR Systems Center, chat among the exhibitor booths.

photographs taken by Randy Schreifels and Kathy Gurchiek

ABOVE: Army liaison Lt. Col. John A. "Drew" Hamilton Jr., center, peruses some of the hot sellers at the Barnes & Noble booth.

BELOW: The Bar J Wranglers of Teton, Wyo. regaled those at the chuck wagon-style dinner with western music and tall tales.

ABOVE: Mary Alice Watson, a test analyst for Boeing Information Services, takes a break to read Mary Higgins Clark.

ABOVE: John Smith, center, Systems Engineering for Lockheed Martin, works with computer equipment at Lockheed's booth.

RIGHT: *CROSSTALK* publisher Reuel "Rudy" Alder, left, honors Robert Martin of MITRE as one of the journal's top 10 authors for 1998.

Earned Value Project Management . . . an Introduction

Quentin W. Fleming and Joel M. Koppelman
Primavera Systems Inc.

Earned value is a project management technique that is emerging as a valuable tool in the management of all projects, software projects in particular. In its simplest form, earned value equates to fundamental project management. Here the authors describe the technique in a storybook form. It is not necessarily a true story . . . but it could be.

ONCE UPON A TIME there was a young man who wanted to be a project manager. Don't ask us why.

In school the young man took the most challenging of the technical subjects, but he also liked to manage things. He graduated with a master's degree in a technical discipline and immediately went to work for a small but fast-growing high-tech company. This company was a leader in developing new products for its niche of the market. The company had just gone public and its initial public offering of stock was a huge financial success. He knew he had joined the right company. All he wanted was his chance at bat. He wanted to be a project manager.

A year went by and he had yet to receive an assignment of any consequence. He was becoming discouraged. He considered updating his resume to start looking around. If his present employer did not recognize his talents, perhaps others would. He did not have time to waste.

One day as he walked down the hall the chief executive approached him. She inquired as to how he was getting along. Then she asked him, *"How would you like an important assignment as manager of a development project?"* The young man could hardly convey his enthusiasm. Then the CEO said, *"If you are interested, call my secretary and get on my calendar for the first thing in the morning."* As she left, she commented to him, *"This is an extremely important project for the company, and I think you could manage it nicely. See you then."*

Our young man got little sleep that night. Imagine, his chance to manage a project — to be a project manager. He was in the chief executive's office 30 minutes before she arrived. When they met she started by saying, *"This is one of the most important potential new products we have in the pipeline, but it needs some innovative thinking, and that is why I think you would be the right person to take this on. I need fresh ideas incorporated into this product."*

She outlined the concept for the new product. It was exactly the type of work he had prepared himself to do. She asked him to gather a half dozen cross-functional people from within the company and to prepare a project plan for her approval."*If you have any problem getting people, use my name to break them loose. I don't want stonewalling by anyone; this product is important to our future growth."*

Then she closed the meeting by saying, *"The time to market is most critical on this project; I know others are working on it, and I want to be first into the marketplace."* The young man got the message, and it was better than he had ever hoped. On his way out she mentioned another issue.

"I would also like you to use a technique I have heard about but cannot seem to get started here: earned value management. Have you ever heard of it?"

"Yes, of course. We studied it in school and I think it would work well on this project," he replied.

"Good. I look forward to seeing your performance plan," she told him.

The young man circulated within the company and got commitment from the right people to do the planning. This was a young start-up company so the "brick walls" so pervasive in older, more established companies had not set in. All he had to do was mention that the boss was behind this assignment and he got his people. He did not even have to describe the details of the assignment, they all knew it was high priority.

Planning for Performance Measurement

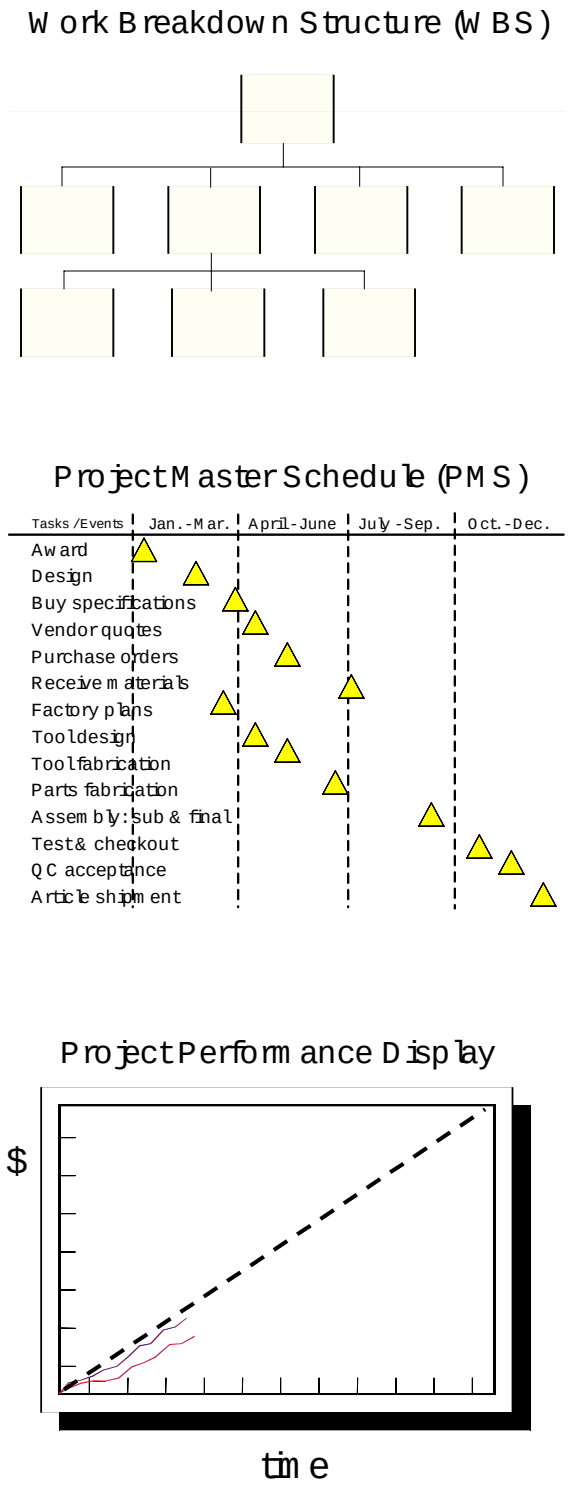
His team met at his apartment to prevent interruptions and phone calls. *"It shouldn't take us very long to put a plan on paper,"* was his opening remark. They spent the day conceptualizing and defining the project. After he solicited the team's ideas, he planned to prepare the final plan for review and approval of the team, prior to submittal to the CEO. The project manager wanted everyone to buy into the project plan. They all knew exactly what was required in order to employ earned value performance measurement. It was classic "Project Management 101."

First they had to define what constituted 100 percent of the assumed project scope. They used a Work Breakdown Structure (WBS) diagram. Next they would decompose the project scope into measurable tasks, each with an estimated value, and assign responsibility for actual performance to some functional manager within the company. They used a WBS dictionary to record their thoughts. They knew that their project had 10 units to develop and test, and that each unit would require about the same level of resources to accomplish.

Next they would take the work, broadly conceptualized from the WBS diagram and dictionary, and prepare a detailed plan and schedule for all the major critical tasks. After a few iterations they had their Project Master Schedule (PMS), fully supported by critical path methodology. They did a forward and backward schedule pass to provide assurances that their PMS was viable. The project would take 18 months to perform from go-ahead to completion.

Lastly they estimated the resources required to produce these 10 units, which constituted the total project. Each article would cost $150,000 to produce, thus the total project would run $1.5 million dollars to complete. They charted their requirements as illustrated in Figure 1, which they termed their project management plan. This display would contain the three critical elements of the plan: WBS, PMS, and a project performance display graph. Each element was supported by detailed break-outs. This process is typically called bottoms-up planning. The team had done its job; it was now time for the project manager to take its plan to the CEO for her approval.

Figure 1. *Project management plan.*



Management's Approval

The project manager made a copy of the project management plan and gave it to the CEO's secretary so the CEO could review it prior to the approval meeting. When he was at last able to meet with the CEO, it was obvious that she had thoroughly read the entire plan; everything was marked and color coded. He hoped she liked what she had read.

The CEO opened on a positive note. *"This is the finest internal project management plan I have ever seen as head of this company, and we will use it as a model for all our future projects to follow."* The project manager was off to a good start.

"However, you must not have heard parts of my requirements. Time to market is most critical on this project, and you are projecting a casual schedule of 18 months. That is completely unacceptable. I need this project completed in not more that 12 months, can you handle that?" The young man took a deep breath.

"Of course we can," he said. He had no clue as to how he would do this, but the message from on high was becoming pretty clear.

"Also, I think you have gold-plated this job at a cost of $1.5 million, that also is unacceptable!" The boss was relentless. *"The most I could allocate for this project would be $1 million; we are not a big company, I have other commitments. Can you handle that?"*

The young project manager was beginning to understand why she had become CEO at such an early age . . . she was one tough person to deal with. Without hesitation the young man accepted the budget dictate.

The CEO realized that she had come down pretty hard on the young man and wanted to provide some consoling words before he left.

"Again I want to emphasize that this is the best project plan I have ever seen in this company. It will be our model for others to follow." Her words were some comfort, although the project manager was starting to worry about what he would say to the other members of his team. Their buy-in was essential to him.

As he was leaving the office the CEO said, *"I am very pleased that you are going to employ earned value measurement on this project. I would like to review your performance each quarter, at say three months into your 12-month project."*

"She never lets up," was the thought that raced through his mind. *"What do I tell the others?"*

Welcome to the World of Project Management

Let us stand back from this story and try to assess what took place. A project team met and developed a thorough, comprehensive project plan, with sufficient supporting data and schedule metrics so they could measure their earned value performance from start to completion. In particular, they had scoped

100 percent of the total assumed project before they would begin to perform and created a plan that could be measured. Good.

Their supporting bottoms-up detail indicated that they needed 18 months to complete the project, and the boss directed them to do it in 12 months. They estimated the costs for the project at $1.5 million and the boss cut it to $1 million. What do we call this kind of an environment the young project manager experienced for the first time? We call it real-life project management.

Rarely do we ever get the total time we think we need to reasonably perform the job. We are always competing with others to do something first. The authorized budgets are rarely what we estimate we need to complete any job. We frequently are given what has been termed "a management challenge" and we do our best. It matters not if these management challenges are arbitrary, unreasonable, unattainable, unrealistic, stupid, and so forth. As project managers, we must find a way to get it done.

Welcome to the world of project management.

The First Quarterly Project Status Review

Three months went by. It was time for the team to present its performance results to the chief executive and the management committee. This would be an awesome new experience for the young project team, but working in its favor was the fact that the team was performing to a detailed plan, and knew exactly what it had to do from the go-ahead.

A brief summary of the team's results indicated the following: Three units had been scheduled for completion at the three-months point, but only two were accomplished, thus members were slightly behind their planned schedule. They had forecasted expenditures of $300,000 and had committed $300,000, so they were right on their funding profile. An optimistic person could easily paint a positive picture of this project.

"We are a little behind schedule, we are right on our spend plan; leave us alone and life will be good," would be the spin put on these results by most practitioners.

However, the chief executive had specifically asked that this project employ earned value project management, and that requires a slightly different orientation with these same project performance data. Earned value management requires a detailed, bottoms-up performance plan, measurement taken against one's own plan, and a periodic forecast of the final expected results, based on actual performance results. Earned value requires detailed measurement against the project plan. In order to employ earned value, there must be a plan in place that allows the continuous measure of seven points of data. This may sound complicated and cumbersome, but it is not. It is simply the kind of data most projects have, but it may not be looked at in quite the same way. Earned value has a focus on its percent complete position against its (100 percent) defined scope.

In order to employ earned value, we must first know at all times what the planned value is as of any point in time[1]. To determine this we need to focus on two issues.

We must determine (1) how much physical or intellectual work we have scheduled to be completed. This is a direct fallout of those detailed tasks contained in our PMS. (Important point: Earned value requires a master project schedule; without a master project schedule one cannot perform earned value management.) In this case the PMS described three units to be accomplished as of the measurement period.

We need to determine (2) the budgeted value of the work scheduled. We were authorized $100,000 per unit, so our budgeted value for work scheduled was $300,000. Thus, we have set our planned value for the first three months of the project at $300,000[2].

Next we will want to measure our earned value for the reporting period. To measure this we need two new points of data, which we will call items (3) and (4).

As of the reporting period, (3) how much of our scheduled work have we actually accomplished? We examine our PMS and find that we have accomplished two of the three units we originally scheduled.

Next, (4) what is the budgeted value of the work actually performed? In this case we were authorized $100,000 per unit, so our earned value for the reporting period is $200,000. (Never mind actual costs at this point, they will only confuse the issue.) Thus, items three and four constitute our earned value for the period[3].

The next item we need to determine is, for the earned value work we have accomplished, (5) what costs have we actually spent and/or incurred? We look at our cost ledger and find we have incurred actual costs of $300,000.

We now have our earned value results for the first quarter, quantified in dollars, and a performance pattern is starting to emerge:

Planned Value —	*$300,000 (items 1 and 2)*
Earned Value —	*$200,000 (items 3 and 4)*
Actual Costs —	*$300,000 (item 5)*

We now need to ascertain our project performance variances, which is a slightly different look at data with earned value measurement.

We need to understand (6) the schedule variance, which in earned value is the difference between our planned value scheduled and our earned value achieved. In this case, we planned to accomplish $300,000 of work, but only did $200,000, so we are behind our planned schedule by $100,000. Not so bad until we realize that we only accomplished 67 cents for each dollar we planned to do.

Lastly, we need to know (7) what our cost variances have been. This is determined by relating our earned value accomplished against the actual costs spent or incurred. Thus, we spent $300,000 in actual costs to accomplish $200,000 in earned value. Not so good when we realize that for each dollar we spent we got only 67 cents of value earned.

The team put the results of its earned value performance on a display chart for presentation to the management committee, as is illustrated in Figure 2. Not a pretty sight, but one of

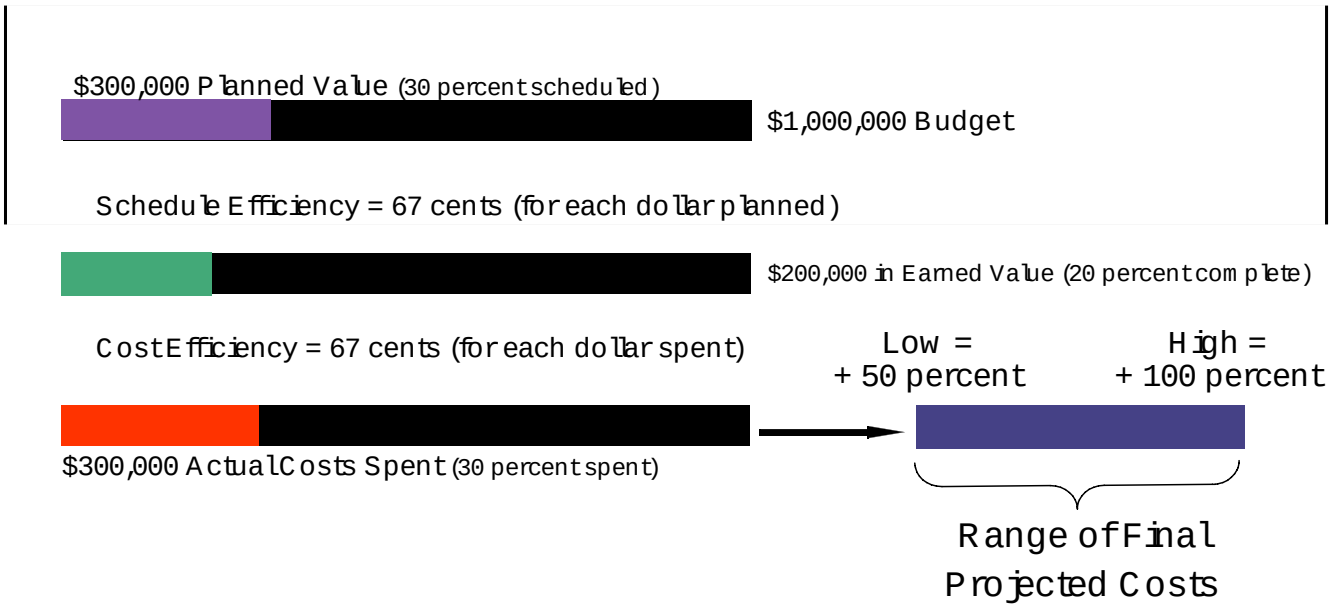


Figure 2. *Earned value performance results.*

extreme importance in the portrayal of the true status of project performance. This project at the end of the first quarter is behind its planned schedule, and is overrunning its costs. At the 20 percent completion point, monitoring earned value data, it is forecasting a significant final overrun.

If the project continued at its present cost efficiency rate of 67 cents for each dollar spent, it would need 50 percent more budget to complete the work ($1,000,000 / .67 equals $1,500,000). If it also tries to get back on the 12-month schedule, it will have to add additional resources to do the same work, so the projected costs would equate to a 100 percent overrun.

Most people do not like to hear bad news. But this chief executive knew that bad news does not improve with time, it only gets worse. At issue: Bad news known at the 20 percent point in a project's lifecycle gives management some opportunity to take corrective actions and alter the final results.

Conversely, bad news that is ignored or not addressed until perhaps the 80 percent completion point severely limits management's opportunities to make the necessary changes to recover performance.

This was exactly the kind of display the CEO wanted to see on this most critical project. She now declared, *"Thank you for this presentation; it has been most informative. I now know I was perhaps a little too arbitrary in my initial budget authorization to you. I will authorize you a revised budget amount of $1.5 million to complete this project."*

"Thank you," was the surprised response from the young project manager. He knew that the team needed at least that amount to complete this project.

(One of the primary reasons earned value results become so reliable at the early phases of a project's lifecycle — at the 15 percent to 20 percent point — rests on the human nature side of the planning process. If one has a period of project performance extending one full cycle, where will you likely place your best planning — in the early periods or in the later periods? Likely in the early periods, and hope for the best in the later periods. Also, if one has a severe budget challenge, where will the most adequate budget be distributed— in the early or late periods of the project? Likely in the early periods. It is human nature to provide the best planning and the best resources to the early periods, and hope for the best. Thus, the results of earned value performance measurement have been found to be most reliable, even at the early periods, say 15 percent, of the lifecycle of a project.)

But the CEO was not going to let anyone off the hook just yet.

"However, I want you to catch up on the late schedule position, and bring us a completed project in another nine months. Can you do that?"

"Yes we can, but it will take an accelerated schedule, and that will likely cost us the full $2 million as we have presented to you," (see Figure 2), was the project manager's reply.

"OK, I will authorize this project a total budget of $1.5 million but ask that you complete it within the 12-month schedule," the CEO directed. *"However, as we both well know, to recover this behind-schedule condition will likely cost us some money, so I will put $500,000 in my management reserve in case we need it. But it is not your money and we want you back on schedule. Am I making myself clear?"* said the CEO.

"Absolutely clear, and we promise to do the best we can for the authorized budget," said the project manager.

"But getting back on schedule is your main performance objective, and the budget goal is simply my management challenge to you. Understand, the schedule comes first," was the CEO final comment.

"*Understood,*" said the young project manager, who was beginning to appreciate the delicate role he was playing.

The Value of Earned Value

Standing back from this situation, we see that this project was likely under-budgeted (at $1 million) from the start. But based on what was authorized and what the project performance was experiencing, the likely final forecast of budget needs was in the statistical range of between $500,000 to $1 million over the official budget. Both the project manager and the CEO clearly understood that fact. But the CEO was not ready to relax her management challenge to this team. She released an additional half a million dollars to the project, but asked that they also get back on schedule. Getting back on schedule would cost additional resources, and likely require the full million dollars to achieve. But she was not ready to authorize the full amount.

This chief executive knew the benefits of employing earned value. She believed the accuracy of data that was being reviewed by the project team and the final projections of required costs. At the 20 percent completion point the team was predicting an overrun of between 50 percent to 100 percent, and she was convinced that this would be the case. In order to fund the completion of this critical project, she took immediate steps to cancel two other internal projects of lesser value to the company. She knew what she had to do in order to fully fund this highest priority project. Other executives who do not employ earned value or do not rely on the performance data often find themselves overly committed in their project portfolios, sometimes experiencing catastrophic results.

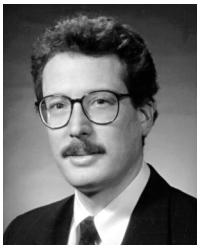
This project was completed on time, within the 12-month schedule, but at a final cost of close to $2 million. The new product worked as hoped, and the additional funds to complete the project were made available by the CEO canceling two other projects of lesser importance to the company.

Life was good at this company, and the young project manager's career was off to a good start. ◆

About the Authors

Quentin W. Fleming, senior staff consultant to Primavera Systems Inc., has more than 30 years industrial project management experience. He held various management assignments with the Northrop Corp. from 1968-91, served on an earned value corporate review team, and wrote the corporate policy directive on scheduling. He was president of the Orange County Project Management Institute (PMI) chapter and developed and taught four PMI Project Management Professional tutorial courses covering scope, cost, time, and procurement management. He has a bachelor's and a master's degree in management and is the author of seven published textbooks, including *Earned Value Project Management*, which he co-wrote with Joel M. Koppelman.

E-mail: QuentinF@Primavera.com



Joel M. Koppelman is president of Primavera Systems Inc., which provides a family of project management software products. Before co-founding Primavera in 1983, he spent more than 12 years planning, designing, and managing major capital projects in the transportation industry, including duties as vice president and chief financial officer for Transportation and Distribution Associates Inc. Before that, he was affiliated with the management consulting firm of Booz Allen Hamilton Inc.

Koppelman is a registered professional engineer with a bachelor's degree in civil engineering from Drexel University and a master's of business administration degree from the Wharton School of the University of Pennsylvania. He is a frequent speaker at universities and for international management organizations.

E-mail: JKoppel@Primavera.com

Notes

1. The Department of Defense (DoD) has called this the Budgeted Costs for Work Scheduled (BCWS) for three decades, but we choose to call it simply the Planned Value.
2. The fact that we originally estimated that each unit would require $150,000 to accomplish is only interesting to us. Management has authorized $100,000 per unit, and does not want to hear about other issues.
3. The DoD typically has called this the Budgeted Costs for Work Performed, or BCWP.

Something from Nothing

Jeremiah Smedra
Software Technology Support Center

Project stakeholders can benefit greatly from implementing a common, defined project management methodology. Far too often, it is the pain of failure and crisis that motivates change. Rather than suffer the ill effects of failure and crisis, we should improve our projects now. This article presents an overview of the Project Management Institute's methodology.

WE SHOULD NOT BE surprised when failure is the fruit of poorly-defined, ill-planned, and arbitrarily executed efforts. In the play "King Lear," Shakespeare writes, "Nothing will come of nothing." Projects are not exempt.

Project management methodology provides the best possible framework for project success. Although all project management environments (software development, construction, defense acquisition, etc.) have unique qualities, they share more commonalties than differences. It does not matter whether you are a practitioner, mid- or senior-level manager. If we are not all dancing the same steps, success is difficult — if not impossible — to achieve. We may get the job done. However, it may not be fun, we may not look good, and we may not be able to call it successful.

Definitions

A project is "a temporary endeavor undertaken to create a unique product or service"[1]. Management generally is concerned with producing key or necessary results. Methodology defines how the project is managed without impacting the effort's uniqueness. If we can define a common project management methodology, communicate it to those we work for and with, and implement it consistently, we will be on the way to reducing risk and actualizing success.

Since 1969, the Project Management Institute (PMI), a nonprofit organization consisting of practitioners and academics, has led the way in researching, organizing, and recording project management methodology. The culmination of its efforts is a work-in-progress, the Project Management Body of Knowledge (PMBOK). The PMBOK model is a structured identification of the skills, concepts and techniques common to the project management field. It is a description of the knowledge and practices commonly found in projects and not a formula to be uniformly applied to all projects. PMI also maintains a professional certification for project managers: the Project Management Professional (PMP). The PMBOK model is widely recognized as the commercially proven and accepted standard for project management. You can download a free copy at http://www.pmi.org/publictn/pmboktoc.htm This article will present a basic outline of PMI's model for project management.

The PMBOK model presents three primary dimensions of project management. These are the lifecycle, the process groups, and the knowledge areas. Each dimension is unique, although related and interdependent.

Life Cycle

The project life cycle is the big picture. The life cycle divides the project into a series of phases which provides better project control. The four common phases in a project life cycle are concept, development, implementation, and termination or closeout.

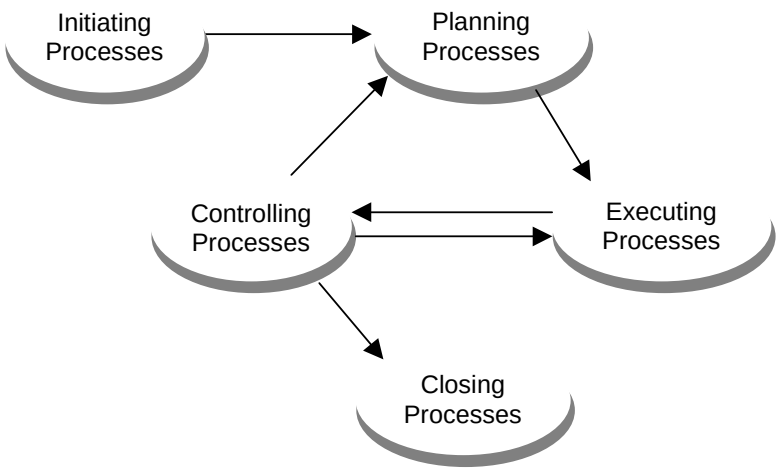
Phases are easy to recognize. The delivery or completion of a major deliverable usually characterizes the end of a phase. For instance, a feasibility study or architectural design might conclude the conceptual phase. The developmental phase might conclude with the project plan. The implementation phase would conclude with the completion of the product or service. Termination might conclude with the customer sign-off, completion of a lessons learned database, and collection of any historical documentation. By treating each phase as a project, we separate complex projects into more manageable pieces.

Process Groups

The five process groups are initiation, planning, executing, controlling, and closing. They are "concerned with describing and organizing the work of the project. ... The process groups are linked by the results they produce — the result or output of one becomes an input to another" [1]. Figure 1 represents how the five processes relate.

The following is an example of what the processes would look like for an actual project. A customer decides he wants to make a change in the product's requirements and the project's production/development is 50 percent complete. Initiation begins with the customer's request. However, the initiating process directs us to document our customer's

Figure 1. Links among process groups [1].



Process	Inputs	Tools and Techniques	Outputs
6.1 Activity Defintion	• Work breakdown structure • Scope statement • Historical information • Constraints • Assumptions	• Decomposition • Templates	• Activity list • Supporting detail • Work breakdown structure updates
6.2 Activity Sequencing	• Activity list • Product description • Mandatory dependency • Discretionary dependencies • External dependencies • Constraints • Assumptions	• Precedence diagramming method (PDM) • Arrow diagramming method (ADM) • Conditional diagramming methods • Network templates	• Project network diagram • Activity list updates
6.3 Activity Duration Estimating	• Activity list • Constraints • Assumption • Resource requirements • Resource capabilities • Historical information	• Expert judgement • Analogous estimating • Simulation	• Activity duration estimates • Basis of estimates • Activity list updates
6.4 Schedule Development	• Project network diagram • Activity duration estimates • Resource requirements • Resource pool description • Calendars • Constraints • Assumptions • Leads and lags	• Mathematical analysis • Duration compression • Simulation • Resource leveling heuristics • Project management software	• Project schedule • Supporting detail • Schedule management plan • Resource requirement updates
6.5 Schedule Control	• Project schedule • Performance reports • Change requests • Schedule management plan	• Schedule change control system • Performance measurement • Additional planning • Project management software	• Schedule updates • Corrective action • Lessons learned

Figure 2. Time management [1].

request, perform an impact analysis, and meet with the customer to exchange expectations.

During planning, we revise schedules, budgets, and other related documents (scope, work breakdown structure, and risks) that make the new requirement part of the project. In this example, executing and controlling are not significantly impacted by our requirements change. The product is built based on our new set of requirements, and measures would be used to direct corrective action as needed. Closing ensures that the customer's request to change requirements was documented to conclusion.

Project Processes/Knowledge Areas

At its most detailed level, PMBOK defines nine unique but often overlapping project areas. The four primary knowledge areas are scope, time, cost, and quality. The four facilitating areas are human resources, risk, procurement, and communication. The remaining overarching area is integration. Integration is concerned with properly coordinating the other knowledge areas. Each knowledge area consists of a series of processes.

Each process includes inputs, tools and techniques, and outputs. (Figure 2.) Inputs are the items needed to complete the process. The tools and techniques define what to do to the inputs. The outputs define the product(s) of the process. The PMBOK guide defines and explains the items listed in the inputs, tools and techniques, and outputs.

Nothing from Nothing?

Although project management can be easily defined, it often looks like fire fighting. Managers and team members act like smoke jumpers. They parachute in on the hot spots, beat back the fire to a reasonable slow burn, and race off to fight the next flare-up. This method of project management is rarely successful. We expect something from nothing.

Devoting resources to training, process improvement, and other similar efforts can be difficult to justify. It often feels and looks as if we are neglecting our primary responsibilities. However, these efforts pave the way for more efficient and effective work. Projects will benefit from the organization, attention to detail, and common language of the PMBOK framework. Rather than continuing to expect results without a firm methodology, we should contribute to teaching and implementing a proven framework for project success. ◆

About the Author

Jeremiah Smedra is a consultant with the Software Technology Support Center. He provides support to Department of Defense organizations pursuing process improvement and project management education. He has a bachelor's degree in marine engineering systems from the Merchant Marine Academy in Kings Point, N.Y. He is a registered Engineer-in-Training (EIT) and a PMI-certified Project Management Professional.

Software Technology Support Center
Ogden ALC/TISE
7278 Fourth Street
Hill AFB, Utah 84056
Voice: 801-777-7230 DSN 777-7230
Fax: 801-777-8069 DSN 777-8069
E-mail: smedraj@software.hill.af.mil

References

1. Project Management Institute, *A Guide to the Project Management Body of Knowledge*, Project Management Institute, Upper Darby, Pa., 1996.
2. Adams, John R., et.al., *Principles of Project Management*, Project Management Institute, Upper Darby, Pa., 1997.

PROJECT MANAGEMENT SERVICES

The Software Technology Support Center offers a number of useful PM services to organizations within the Air Force, other government agencies, and their supporting contractors. Services include:

- PMBOK area specific workshops
- Tool training
- Executive sessions
- Consultation and mentoring

All services are provided on a cost-recovery basis. To get your project management effort off to a good start or give it a shot in the arm, contact the author for further information regarding PMI and the PMBOK model.

Managing Risk Management

August C. Neitzel Jr.
National Reconnaissance Office

This article will address the development of a pilot risk management effort within the National Reconnaissance Office's Imagery Intelligence Systems Acquisition and Operations Directorate (IMINT). The topics to be covered will be the background and rationale for the instantiation of a risk management program and the working relationship with the Software Engineering Institute in tailoring its processes that led to the development of an automated Risk Management Tool. The methodologies and processes in place, as well as lessons learned and future follow-on efforts also will be addressed.

Background

The National Reconnaissance Office (NRO) underwent a consolidation and collocation of its resources to northern Virginia from late 1993 through early 1994. This brought together, for the first time on a large-scale, members of the NRO uniformed services and the Central Intelligence Agency.

In March and April 1996, the director of the NRO commissioned a Baldridge study to assess the quality of life and the processes in place in the NRO. The study addressed a broad spectrum of topics. The results indicated issues existed in the acquisition and planning processes, communications, and personnel. These issues were, to a large extent, due to cultural differences of the newly combined military and civilian organizations.

While other elements of the NRO addressed the wider NRO Baldridge issues of communication and personnel, the NRO's IMINT focused on its internal acquisition and planning processes. To facilitate this focus, IMINT requested that the Software Engineering Institute (SEI) from Carnegie Mellon University, a federally funded research and development center (FFRDC), conduct its Software Acquisition-Capability Maturity Model[SM] (SA-CMM) [1] survey of IMINT. IMINT's goal was to achieve an overall improvement in its acquisition processes.

Starting in August 1996 the SEI conducted the IMINT SA-CMM. The SA-CMM survey allowed the SEI to interview a broad cross-section of IMINT's government and contractor (i.e. development, FFRDC, Contractor Advisory and Assistance Services, and System Engineering Technical Assistance) personnel. The results of the survey and the Baldridge study were fairly consistent in the area of process improvement.

Although the SEI SA-CMM survey identified many strong acquisition process areas (e.g. rigorous configuration management, development standards, and acquisition methodology) it found weaknesses in the uniform application of the established processes to the acquisition of NRO's systems. Risk management was a notably weak area. In this case the government program office had no documented processes to follow. This was in stark contrast to IMINT's contractor community, which in general had very proactive and rigorous risk management programs in place.

The briefing to IMINT management by the SEI SA-CMM team concluded that IMINT should embark on an acquisition improvement program, with an emphasis on establishing a Team Risk Management (TRM) program. More specifically, the SA-CMM team recommended forming a pilot TRM program. IMINT management adopted the recommendation.

IMINT management's rationales for needing a strong risk management discipline are the same as those shared by most of their Department of Defense (DoD) and industrial mission partners. As systems become more complex and interactive, it is essential to identify and understand the interrelationship of the risks within and across programs. The programs must appreciate how a risk in one element may cause a risk in another element. Risks that are not proactively managed eventually begin to manage you. Early risk assessment and mitigation can and will minimize downstream surprises and problems. Shrinking budgets and tighter schedules virtually eliminate any margins that could be retained to offset problems that might occur late in a program.

Following the SEI SA-CMM recommendation, IMINT management selected its command and control development (CCD) effort, for which the author is the program manager, as the vehicle for the pilot TRM program. This selection was made in part because the CCD effort is the most software-intensive acquisition program within IMINT and the NRO, and in part because there was some degree of belief that the SA-CMM process was primarily applicable to software development efforts. The CCD acquisition consists of several million lines of code (new, modified, and reuse) and utilizes C++ object-oriented design (OOD). It is commercial-off-the-shelf products (COTS) intensive and is a large distributed client/server architecture of several hundred servers and workstations. It has multiple deliveries spanning more than three years and over geographically dispersed facilities. In addition to the software sizing aspects of the CCD effort, there was some degree of the "let Mikey try it" syndrome in IMINT's decision. The author being viewed as the resident skeptic, IMINT management seemed to think that if CCD bought into the TRM process, others would readily follow. On this ceremonious note the pilot program was off and running.

Software Acquisition-Capability Maturity Model (SA-CMM) is a service mark of Carnegie Mellon University.

The Pilot Team Risk Management Program

The first step was to reconvene a SEI/contractor/government team and establish a plan of attack. CCD elected to initially limit the scope of the pilot program to a subset of their overall acquisition activities. The CCD acquisition effort had several incremental deliveries in its plan. One of the later deliveries was selected as the basis for the pilot effort. This later delivery involved one of our subcontractors who was chosen to be the primary participant in the study, with our prime contractor providing a supporting role. The driving rationales for this were multifaceted. The main one was to minimize any potential disruption to more time-critical activities. Another was to select an activity early enough in its acquisition process that it might better accommodate any potential change. A third was to select an activity where the cultural differences were the most noticeable.

CCD initiated its SEI-led Software Risk Evaluations (SRE) in January 1997. The CCD contractor was chosen to begin the process and conducted its own, separate Risk Identification and Analysis (RI&A) and Mitigation Strategy Planning (MSP) phases in two five-day periods concluding in March 1997. The CCD government team immediately followed with its own SRE RI&A and MSP phases in April and May 1997. The contractor and government SREs were done separately to ensure confidentially and to build a baseline of risks to be selected by both organizations for joint mitigation in a TRM environment.

The CCD program office's RI&A portion of the SRE involved four independent teams. Members of CCD technical staff (i.e. area managers) made up team one, CCD management made up team two, members of CCD's Aerospace FFRDC cadre made up team three, and members of CCD's operational customers and systems integration contractor made up team four.

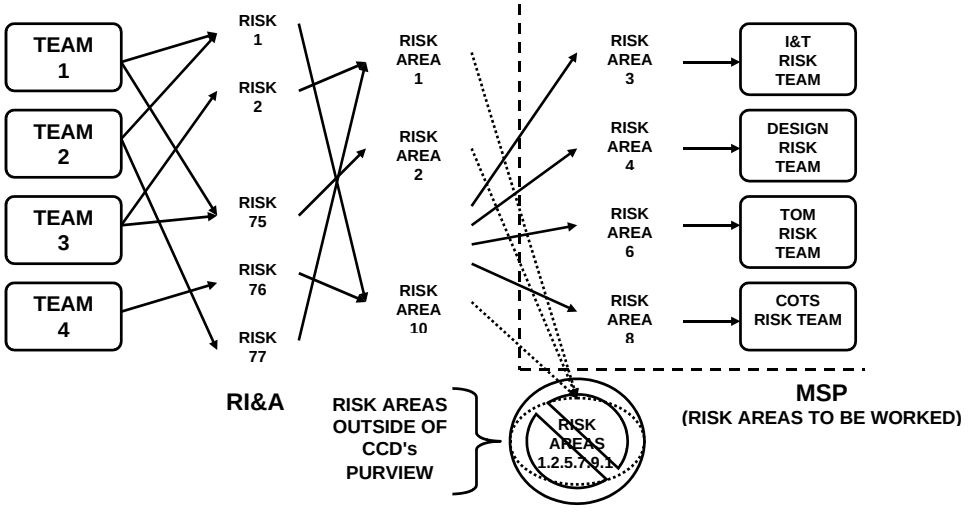
Each of the four RI&A teams utilized the SEI SRE taxonomy questionnaire. The four teams generated 77 risk statements. In some instances a risk statement was unique to a team. In other cases, multiple teams generated the same risk statement. SEI compiled and tabulated the 77 statements and assigned them into 10 risk areas or affinity groups. The 10 areas and the number of risk statements generated within each were:

- *Risk Area 1* — Requirements (11)
- *Risk Area 2* — Staffing (7)
- *Risk Area 3* — Integration and Test (I&T) (7)
- *Risk Area 4* — Design (8)
- *Risk Area 5* — Schedule (3)
- *Risk Area 6* — Transition to Operations and Maintenance (TOM) (7)
- *Risk Area 7* — Program Office Management (16)
- *Risk Area 8* — Commercial-off-the-Shelf products (7)
- *Risk Area 9* — Prime/Subcontractor Relationships (4)
- *Risk Area 10* — Contract Management (7)

The joint TRM process commenced in June 1997 with a government/contractor/SEI MSP session. The joint team chose to pursue Risk Areas 3, 4, 6, and 8 for mitigation. A risk team was assigned each of the selected risk areas for further characterization and mitigation strategy development. It was thought that these four areas would provide more than enough risks to attempt to mitigate in a pilot program. In addition, it was thought that the other risk areas (1, 2, 5, 7, 9, and 10) fell outside the purview of the CCD team and the probability of successfully mitigating any of the associated risks was low and of minimal payback. For instance, in the area of requirements, most of the requirements' instability risks were driven by external elements to either CCD or IMINT. The likelihood that the CCD team could unilaterally control the flow of changes was improbable. Interestingly though, these areas subsequently were assigned and worked at a higher management level when the CCD risk management process was adopted at the IMINT program development level. Figure 1 provides a representation of the RI&A and MSP process CCD followed.

The area of risk training was a key aspect in the development of the CCD pilot TRM program. The CCD team took advantage of the SEI risk training that stepped us through the SRE RI&A and MSP, Continuous Risk Management (CRM), and TRM concepts. However, we elected to skip the risk clinic training SEI offered. We thought (incorrectly) that the details taught in the clinic were unnecessary and we already knew what we needed to know to succeed. As we progressed through the various risk management stages and attempted to develop our pilot plan, we soon came to the conclusion that the risk clinic was a valuable tool we should not have been so cavalier in discarding. The team found it was having difficulty with not only the risk management lexicon but also in developing a firm understanding of what differentiated a risk from an issue/problem. With our belated participation in the risk clinic, we discovered that the team members inherently understood the steps each was tak-

Figure 1. Software risk evaluation process.



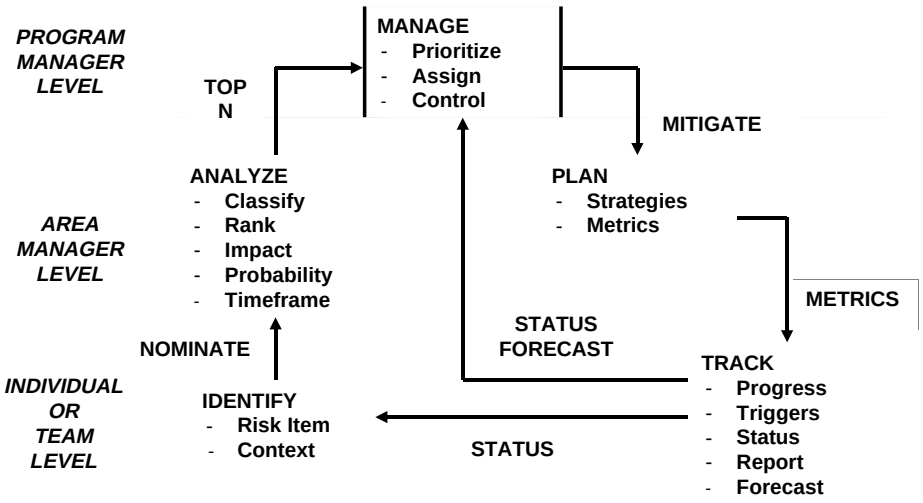


Figure 2. Continuous risk management flow.

ing to identify, quantify, and mitigate risks. The problem was in establishing a documented and uniform process that the entire team could follow. We utilized the CRM flow concept that is documented in SEI CRM handbook [2] and tailored it to fit our process flow.

In the CRM process we developed for our pilot program, we allocated responsibility for the initial identification of a risk to the teams and individuals most readily familiar with the program element. It is the function of these individuals/teams to define the risk item and put it in a context that clearly categorizes it.

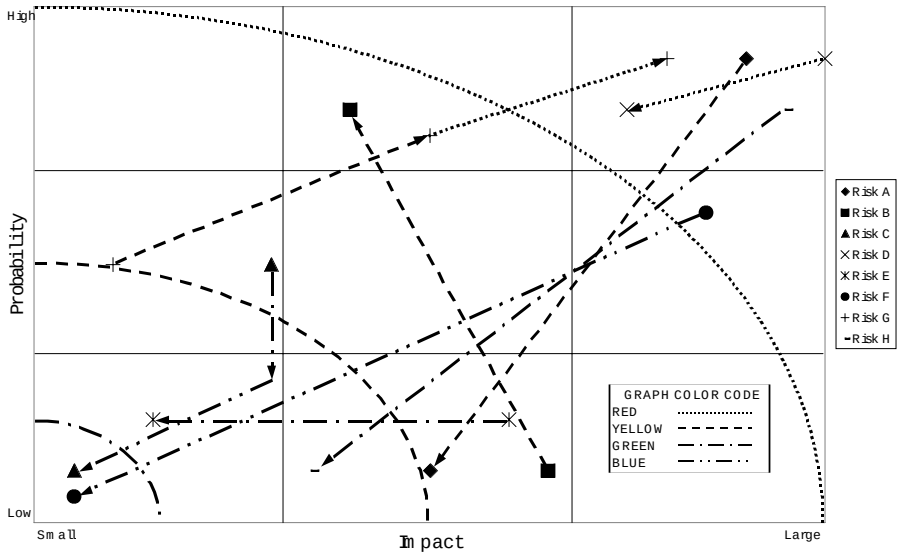
These risks are passed on to the CCD area managers, who analyze them to determine the potential impact, probability, and timeframe of occurrence. The area managers then proceed to classify the risks according to impacted area, closure criteria, decision timeframe, and response. In our adaptation of the CRM flow, we added "support" to the existing responses of watch, accept, and mitigate. There are numerous instances where an IMINT risk is present for which CCD would have no mitigation responsibility, but where CCD support would be needed for formulating an adequate mitigation plan. If the area managers chose to accept the nominated risk, they rank its significance relative to all the risks under their purview and pass the top N to CCD management for ultimate prioritization, assignment, and control (i.e. disposition).

CCD management then has the option of modifying any of the risk parameters (e.g. probability, decision timeframe, and impact) and placing the risk in the CRM plan. Once in the CRM plan, the appropriate mitigation strategies are developed along with the metrics needed to assess progress against the plan. The tracking system allows for routine progress and status reports to be generated, as well as producing briefing material to identify current status and forecast future movement. Trigger points are established to alert management and the risk manager of key decision dates or activities for the risk in question. Figure 2 shows a top-level representation of the CCD risk management flow.

The team developed two significant risk-reporting presentations used in briefing senior IMINT management. The first report is a barometric-like representation that tracks our risks throughout the impact — probability continuum. This gives senior management a snapshot of where risks have been and where they are going at a top level. Figure 3 provides an example of our barometric chart. The curved lines that connect the impact and probability axes provide a quick visual assessment of the risk groupings.

The second report, which is still a work in progress, assesses the exposure the program faces on any given risk. This report melds the risks' impact and probability values along with the decision timeframe, budgetary, and Technical Performance Measurands (TPM) factors for a visualization of the risk population's relative exposure. TPMs are a measurement of those items that the NRO has committed to provide its customers. For example, given two risks with equal impact, probability, and decision timeframe, the one that is unbudgeted and adversely affects a TPM probably deserves more management attention than one that is budgeted and has no impact on a TPM. Figure 4 shows an example of the prototype exposure report. In this example, Risk F is ready to be closed, and CCD's second highest priority risk, Risk A, has lower exposure than the next highest exposure risk, Risk G. The implication is that the next level of management probably needs to apply more attention to Risk G than Risk A. In practice, Risk G might fall into a "watch" or "support" category for CCD but into the "mitigate"

Figure 3. Risk barometric chart.



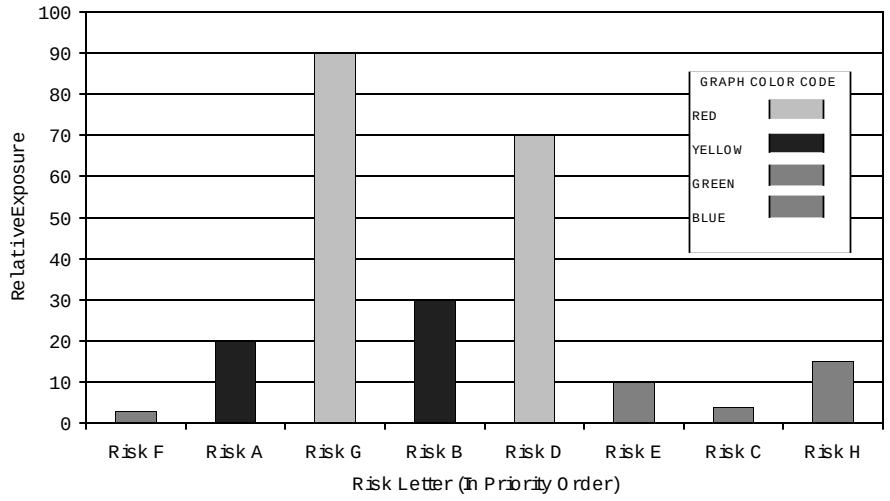


Figure 4. Risk exposure.

category for IMINT at large.

(Note: Figures 3 and 4 are typically represented in a four-color format with red representing items with the greatest risk and exposure through blue for those that are of the least risk and exposure and ready for closure. The figures include a Graph Color Code key for identifying the color scheme in the black and white figures.)

In conjunction with formalizing and documenting the risk process flow, we also established dedicated meetings with formal agendas to nominate and disposition risks within CCD. Each Monday the area managers review and status the risks they are managing. Monthly, at our joint Team Risk Reviews (TRRs) with our contractors, and facilitated by SEI, new risks are nominated, mitigation plans are developed, and old risks dispositioned. We have found it extremely beneficial to have a broad government/contractor/SEI experience base at these TRRs, as it produces a superior mitigation plan.

The Risk Management Tool

We continued to refine and enhance our processes as the CCD pilot risk team progressed through the various phases of the SEI process. One of the more significant products was the development of our Risk Management Tool (RMT).

The RMT is the result of a collaborative team effort between the CCD, government, Lockheed Martin Corp., and ORACLE.

The team's objective was to model the SEI/CCD CRM process established during CCD's risk clinic and to develop an automated interactive Web-based tool — the RMT.

The RMT facilitates a hierarchical approach to propagate risks through the system by enforcing workflow via defined roles and responsibilities for all users. The RMT's assignment feature provides users with the capability to communicate with other users in the system and to move risks through the approval processes. Personnel is notified of risk assignments via automatically generated e-mail. Personnel associated with a risk also is notified via system-generated e-mail when key data items are added or updated.

The RMT's built-in security features provide data protection and partitioning that prevents unauthorized access and enforces the defined hierarchical workflow.

The tool engages the end user with its intuitive graphical user interface (GUI). GUI features include JavaScript-assisted pop-up lists, pull-down menus, and free-form data entry fields. JavaScript also is employed to perform client-side validation of user entries. The user-friendly RMT includes detailed online help and real-time validation checking. Numerous custom query screens and reports provide valuable information on risk status and progress measurement to support decision making. Reports are provided in either textual or graphical format, including the barometric and exposure reports discussed earlier.

The tool is designed for use with a risk-management methodology modeled after the SEI process. When used in conjunction with other established program management processes such as earned value management and critical path methodology, it greatly enhances insight into the acquisition process for program management.

Success Stories

The pilot TRM program developed by IMINT CCD has been successful and forms the basis for the larger TRM program that spans all the acquisition activities within IMINT. The CCD processes provided the foundation for the acquisition activities' Executive System Risk Team (ESRT), which convenes monthly and is chaired by the program director. This forum assesses the most significant risks facing the program and concentrates on the interdependent risks. Many of the risks that CCD identified in the RI&A phase of its pilot program, which were out ofits mitigation purview, now are managed within the ESRT.

In developing the TRM process and propagating its use across the various development disciplines, we refuted the concept that the SA-CMM methodology is limited to software acquisition programs. The "S" in SA-CMM might more accurately stand for "systems" as opposed to "software".

Work is under way to expand the risk program into IMINT's operational elements, although operational personnel do support the ESRT.

CCD has been asked to share its TRM experiences and lessons learned with the NRO's Acquisition Steering Group and Signals Intelligence Acquisition and Operations Directorate (SIGINT) to aid them in the development of their own TRM efforts. Additionally, SEI and CCD have worked with the NRO's Acquisition Center of Excellence to promulgate a TRM concept across the larger NRO community.

A contractor for one of the NRO's biggest customers, The National Imagery and Mapping Agency (NIMA), has asked to utilize the processes that CCD developed in formulating its TRM program. On a more basic level, the TRM program is proving to be of greater and greater utility as IMINT's programs progress through the acquisition phases and near its operational readiness milestones. The

formalization of the risk process has helped to develop a higher confidence level for senior management. They now have better access to and greater insight into the interrelationships of the key development activities. As each of the interlocking development programs have embraced a TRM process, a clearer picture has materialized that shows how tightly coupled these activities are. Not only has senior management's visibility into previously obscure details improved, but other contracting officer's technical representatives within the program have a better appreciation of how risks within its sphere of influence might impact others in very subtle ways.

The development of the TRM program has provided a mechanism for early risk identification and mitigation. This proactive approach allows IMINT to place its risks in better perspective and to focus on those with the highest potential (i.e. greatest exposure) to negatively impact the programs' process. By thoroughly defining and quantifying a risk's potential impact, it has been possible to establish budgetary liens that have withstood detailed scrutiny.

A side benefit is that the government/contractor team has forged a much closer and candid working relationship. The ability to bring together key talents and a broad experience base from the combined government and industry sides of the acquisition process has enhanced both participants.

Lessons Learned

The first lesson that all the participants quickly became aware of was that we should not have bypassed the SEI risk clinic. Although the team inherently understood the basic risk identification thought processes, it was essential that we develop a common lexicon and work through the risk identification formality. The TRM plan and risk process flow that resulted from our participation in the clinic allowed us to further enhance our processes as management requirements have changed.

Some in the organization still treat a risk as a four-letter word. The key is that risks are a natural byproduct of any activity. The more complex and challenging the effort, the greater the inherent risks. Managers need to recognize this and not hesitate in bringing risks forward to senior management. Likewise, senior management should not "shoot the messenger," nor should senior management be over-eager to help. Intervention is likely to restrict the open flow of information.

Differentiating a risk from a problem is still difficult for many. It is essential in the TRM process to identify potential problems and bring them to light as soon as practical. To do otherwise is unproductive. The exchange of information is severely restricted and the ability to develop comprehensive mitigation plans is inhibited.

For the TRM process to work, senior management must buy into the process. It is essential that the management team devotes the necessary time and energy to the process and continually reinforces the required discipline.

The establishment and execution of a CRM process requires a reasonable expenditure of resources. The CCD team spent many hours establishing its process and developing its risk database. The effort needed to maintain the momentum is considerably less, but by no means zero. Our weekly area manager meetings and monthly TRR and ESRT meetings continue to require support to be viable.

Lastly, as the team progressed through the process, we realized that risk management does not stop when an element is transitioned to operations. It is important that operational risks also are managed. In keeping with this recognition, our Integrated Development and Maintenance Organization (IDMO) instituted a risk management process that helps to better focus and prioritize available resources. Our IDMO is actively represented on our TRRs and ESRTs.

The Future

The challenge from NRO management to the team is to quantify the successes that a proactive TRM program can bring to an organization. Although both the CCD team and now the IMINT programs team can point to clear examples of where the risk program has helped identify and mitigate risks, we have not yet established a set of metrics that allows us to quantitatively represent the successes.

The risk barometric graphic (Figure 3) has been very useful in quantifying the progress on any individual risk from inception through retirement, but in itself is not adequate.

The CCD team is investigating the utility of tracking a risk's exposure as a function of time to see if this, coupled with the barometric representation, provides any additional insight. As we continue to enhance our data collection and reporting in this arena we hope that it will address the challenge we have been given. ◆

References

1. Software Acquisition-Capability Maturity Model (SA-CMM) V1.01 (CMU/SEI-96-TR020)
2. *Continuous Risk Management Guidebook*, CMU/SEI, October 1996.

About the Author

August Neitzel earned a master of science degree in electrical engineering from Drexel University after completing a tour of duty with the Air Force. He is a member of Eta Kappa Nu and the Institute of Electrical and Electronic Engineers. He joined the Central Intelligence Agency (CIA) in 1975. In 1982, he began working for the NRO. His career there has spanned the SIGINT program and virtually all aspects of the IMINT program. He is chief of IMINT's command and control acquisition effort and the contracting officer's technical representative for the command and control acquisition contract. He is certified as a Level III COTR by the CIA. Neitzel received the Intelligence Commendation Medal for his work with the CIA.

Voice: 703-435-7217
Fax: 703-808-2038
E-mail: neitzela@nro.mil

Operation Data Storm: Winning the Interoperability War through Data Element Standardization

Mary Linda Polydys
National Defense University

Implementing standard data elements is one of the keys toward effective interoperability. Even though the Department of Defense (DoD) has implemented an active data standardization program, barriers continue to impede standard data element use, thus impacting DoD's vision of interoperability and information superiority. This article addresses barriers in implementing data element standards and recommends several actions that can overcome those barriers. The article also proposes an intensively focused initiative, Operation Data Storm, to increase management attention on the importance of data standards and to resuscitate the data element standardization efforts.

The Plight and Recommendation

Sophisticated technology and systems provide today's warfighter with enhanced capabilities designed to perform assigned peace-time and war-time missions. Many, if not all, of these systems rely on software and computer systems to provide and enhance superior performance. Increased interconnections and reliance on data exchanges supporting readiness demand interoperability. Effective interoperability between systems, including weapon, command and control, combat support, messaging, and automated information systems, is an imperative in achieving information superiority [1].

Data element standardization can provide that interoperability. Standard data elements in software intensive systems provide the coordinated means to describe and exchange data, improve communication, minimize the requirement for data translation software and devices, and eliminate redundant data across the battlefield and functional areas. For example, combatants in a joint task force share standardized location and other information to support the overall mission. Sharing of critical location data, made possible through data standardization, enables synchronization of forces. This example shows that interoperability is a key component of readiness and combat effectiveness. We see that data elements define information across a variety of DoD systems supporting readiness. Standard data elements, therefore, have become as fundamental to readiness as ammunition or fuel [1].

This said, we would think that DoD is well on its way toward implementing standard data elements. True, DoD has been successful in creating standard data elements. For that matter there are 18,000 or more data element standards available for implementation. However, success in implementing those data element standards is less than notable. This is due to both a timid approach in implementing standards, and barriers program and software managers face in using DoD standard data elements. These barriers range from resource availability to commercial-off-the-shelf (COTS) use.

Operation Data Storm is proposed as an initiative that aggressively focuses on implementing data element standards to win the interoperability war and to help deal with the barriers facing program and software managers.

The Resource Barrier

The first barrier impeding data standards use is the availability of adequate resources. Although DoD established an organization to create, manage, and implement standard data[1], most of the resources to date have been focused on policy development and creating data element standards. In other words, the adequacy of resources for policy and creation is sufficient, but the adequacy of resources for implementation is insufficient.

What appears to be missing in this resource formula is a well-formed data administration structure that supports the implementation of data element standards during software development. There are several reasons for providing direct and independent support to software developers. First, software developers are not necessarily skilled in the acquisition and use of standard data elements. Second, the primary goal of software developers (i.e. writing workable code) often conflicts with the primary goal of data element standardization (i.e. information exchange). To minimize the effects of conflicts, systems data administrators can be added to the resource formula.

The systems data administrator is appointed for a single system or for a family of information systems. In addition, the systems data administrator supports the software developer in the acquisition of data element standards and resolves conflicts that arise in trying to use standards. At the same time, the administrator is accountable to the DoD, functional, and agency data administrators for measuring and reporting usage of data element standards and associated improvements in interoperability.

The Cultural Barrier

The second barrier in implementing data element standards is cultural attitudes that negatively impact decisions to use standard data elements. As an example, software developers may display a "not invented here" syndrome. Because soft-

ware developers traditionally create their own data elements, the idea of using someone else's notion of a data element can be considered intrusive. Additionally, software developers may deem that "it is too hard" to implement standards. The data elements software developers create often are tightly coupled to the application domain, making it easier to code the application. It is not always easy to use DoD standards that are domain independent.

Another attitude that may impact negatively on implementing data standards is the notion of "what is in it for me?" What is in it for program and software managers is their satisfaction in contributing to DoD's quest for achieving their vision of information superiority. But this satisfaction can be elusive in the face of near-term schedule and cost limitations. Program and software managers who face these limitations may consider the implementation of data element standards as one of those costs they can't afford. There are no rewards for missing a schedule or exceeding costs just to implement or propose new standards. Besides, there is likely to be little impact on the manager's future for not implementing standard data elements. After all, there are no strict measures of performance or consistent enforcement for ensuring standard data element use.

Lastly, the notion of institutionalizing a data administration community at the software development level, in an environment of scarce resources and downsizing, guarantees resistance from all levels. Managers may view this as the growth of additional bureaucracy that will add little value and cost additional resources that otherwise could be used more productively in software development.

Although many of these cultural barriers can be mitigated through advanced data administration planning as a part of program planning, additional efforts should be taken to overcome attitudes that impact negatively in using data element standards. These efforts include appropriate education and training, special incentives, and reallocating resources from creating to implementing standards. Resource reallocation minimizes the need for additional funds and personnel. Providing special personal and organizational incentives provides a means to reward individuals and organizations for their standardization efforts. Such incentives are not without precedent. Government organizations have long established these kinds of rewards (e.g. quality management, value engineering, and acquisition streamlining rewards).

Finally, using the budget systems' approval processes can provide other incentives in overcoming cultural barriers. Additional resources can be provided to program and software managers who demonstrate their use of data element standards. On the other hand, program discontinuance or special external assistance is a reward to program and software managers who do not demonstrate appropriate use of data element standards.

The Migration Barrier

A third barrier in implementing data element standards is the existence of legacy data in current mission critical information systems that are likely to survive for some time. It may be too costly, impractical, or impossible to migrate all legacy data to standards.

In these cases, the engineering change proposal system can be used to ensure consideration and use of data element standards. As part of evaluating a change proposal, the systems data administrator completes a thorough data analysis. This includes identifying all legacy data elements that are directly or indirectly impacted by the proposed change, mapping those elements to candidate data element standards, assessing the impact of migrating to standards, and providing a recommendation. In the event that migration of some or all of the legacy elements is not practical, the systems data administrator maintains a record of the mapping in a prominent part of the systems documentation to accommodate the development of potential needed interfaces.

In the case that replacement systems are in the planning, analysis, or implementation stages, migration to data element standards is potentially easier. The unfortunate fact is that many program or software managers do not adequately accommodate for data element standards in their migration plans.

The Interface Barrier

A fourth barrier in implementing data element standards is the notion that building interfaces is all that is needed. In some cases, building standard interfaces may be the most expedient way in which to map to data element standards for interoperability. However, there is a significant cost in doing so. This includes not only the cost of maintaining interfaces that grow exponentially with the increase in information interchanges but also the cost associated with increased complexity.

Complexity impacts the ability of software managers to make changes quickly and efficiently, thus driving up costs. In addition, costs of maintaining a growing number of interfaces take away the scarce resources needed to implement new software. In an Air Force Data Strategy Paper [2], an analysis of Air Mobility Command's investment in interfaces revealed that "80 percent of annual software costs are interface maintenance costs, and 20 percent of annual software costs are core system software expenses…." Although building interfaces produces short-term schedule and cost reductions, these savings ultimately are erased during interface maintenance.

The Commercial-Off-the-Shelf (COTS) Software Barrier

A fifth barrier is the policy of the federal government to rely on the use of commercial items (including COTS software) to satisfy information technology needs. In the case of COTS software, this includes the use of vendor-created data elements. Unless it is the norm for a COTS software product, mandating that the product be changed to incorporate DoD data element standards may effect the status of the product as a commercial item and may be cost prohibitive as well. One way to handle the issue of commercially designed data elements in COTS software is through a strategy of interface management. The DoD's Office of the Assistant Secretary of Defense for Health Affairs issued a memorandum [3] that provides some insight into such a strate-

> "This contract language applies to Commercial Off-The-Shelf (COTS) software purchased to satisfy Military Health Services System (MHSS) functional requirements. It does not apply to development tools such as PowerBuilder, Visual Basic or auxiliary utilities-oriented packages…Congress defined performance measures to assess progress toward information technology goals in the National Defense Authorization Act for Fiscal Year 1995…Section 381 paragraph 1.3.2 of this Act specifically establishes measures for data standardization to include number and percentage of DoD standard data elements that are used in migration systems. This can be done through the actual use of DoD standard data elements or the mapping of nonstandard data to DoD standard data...."

Figure 1. Part of the DoD's Office of the Assistant Secretary of Defense for Health Affairs memorandum.

gy. Part of the content of that memorandum is provided in Figure 1. More specifically, the language recommended for inclusion in solicitations or contracts (Figure 2) provides further insight on how to deal with commercially designed data elements.

Support for Implementation

Two recent activities in DoD provide increased support for implementing standard data elements and overcoming the barriers previously mentioned. One of those activities is in the policy support area and the other is in the software engineering area.

Policy Support

In April 1998, DoD published a manual [4] on data standardization procedures[2]. Chapter 7 of that manual provides guidance on implementing data standards. This chapter provides detail on translating DoD data standards into data elements that can be used in software implementation. Included are descriptions of registering the use of DoD data standards, transforming the logical data model to a physical schema, refining a database schema, and improving DoD data standards during software development.

Software Engineering

For more than a year, DoD has been engineering data standards into reusable reference data sets that can be used in software applications. This initiative is called Shared Data Engineering SHADE[3]. SHADE is a strategy that identifies how to share data resources at the application level. It brings together the disciplines of data administration and database administration to identify data requirements and implement database design in a manner that promotes interoperability. To this end, SHADE engineers have transformed the data specified in the DoD Data Architecture (data model and repository) into database components that can be used in DoD systems.

These database components, which include implemented data standards, are called reference data sets. The data element standards included in reference data sets are primarily the elements that can be represented by static data values[4]. The use of these reference data sets supports data interoperability in that they provide uniform representations of standard data elements for use in mission critical systems. In addition, these reference data sets are designed for use under the Defense Information Infrastructure (DII) Common Operating Environment (COE)[5].

There are several hundred reference data sets that can be downloaded from the SHADE Web site and moved into a software application for immediate reuse. This also is an example of software code reuse in that both the data definition language and the table values can be ported directly into most software databases. This eliminates the need to re-enter hundreds of data values, minimizing data input errors.

Operation Data Storm — Last Thoughts

The DoD procedures and SHADE are the tip of the iceberg. Current results appear to be sporadic, at best, and do not deal with most of the implementation barriers. If DoD is to win the interoperability war, more aggressive steps need to be taken to deal with the barriers to implementing standard data elements. In

Figure 2. Specific language to be used in solicitations and contracts.

> "Prior to final agreement and purchase, the government requires the vendor to provide a data dictionary, which includes the following information for each functional data element in the software: the logical data element name, its definition that describes the meaning and the context of the data element in the system, the domain of the data element (the allowable values), the data type, length and a unit of measure if applicable. The vendor is required to submit this data dictionary using the MHSS Health Import Tool (HITool) which can be obtained from www.hirs.af.mil/mhss/. In addition, the vendor will be required to provide additional information for clarification of the individual data element meaning and context to assist Health Affairs in reporting the National Defense Authorization Act data standardization metric.
>
> The vendor shall describe in its proposal to the government and be able to demonstrate within ___ days of final award the applications capability and flexibility to import and export applicable standard MHSS data defined in the task order to or from external sources directly or through standardized interfaces, front-end or back-end translators or utilities."

implementing standard data elements. In addition, a return on investment for implementing data standards needs to be demonstrated and this return can only be confirmed after implementation of standard data elements.

Operation Data Storm is proposed as an initiative that would focus more aggressively on implementation. Operation Data Storm is a focused and concentrated management initiative. The initiative would require DoD Chief Information Officer support, as well as the support of the DoD Acquisition Executive. The first part of the initiative would cover approximately a six- to-eight month period where selected software intensive systems intensively focus on using data standards. The selected software intensive systems should include a wide variety of systems at various stages in the acquisition lifecycle. Selecting systems at various stages will reveal the different kinds of management issues that arise at each stage. At the end of this first part, an assessment would be made to determine any change in direction.

Resources for this initiative can be reallocated from the resources currently used to create data standards; DoD data administrators could refocus their planning efforts to participate in implementation. Getting their "hands dirty," so to speak, will increase understanding of the barriers and problems program and software managers face in implementing the standards policy makers created. An intensively focused effort can provide lessons learned on overcoming barriers that would benefit both the policy makers and developers, and provide an example for future software developers in using standard data elements.

Lastly, any additional standards that are needed during this implementation effort can be created during software development. This will promote a process of creating standards when they are needed for a just-in-time inventory of standards.

The bottom line is a need for proof — proof that implementing data standards decreases cost and proof that implementing data standards promotes information superiority. Unless a return on investment and improvements in interoperability can be demonstrated by using standard data, software development efforts will continue to avoid using data element standards. Operation Data Storm can help provide that proof. ◆

The views expressed in this article are those of the author and do not reflect the official policy or position of the National Defense University.

About the Author

Mary Linda Polydys is chairwoman of the Systems Acquisition Department, Information Resources Management College, National Defense University. A former contracting officer and DoD data administrator, she leads the development and delivery of acquisition education for Chief Information Officers and other information technology professionals. She has a bachelor's degree in decision sciences and a master's degree in information systems from George Mason University in Fairfax, Va. She is a doctoral student in information technology at George Mason University.

Mary Linda Polydys
National Defense University
Information Resources Management College
Building 62, 300 5th Avenue
Fort McNair
Washington, D.C. 20319-5066
Voice: 202-685-3889 DSN 325-3889
Fax: 202-685-3974 DSN 325-3974
E-mail: polydysm@ndu.edu
Internet: http://www.ndu.edu/irmc

References

1. Defense Information Systems Agency Memo, "Implementing DoD Standard Data Elements," April 1996, http://www-datadmn.itsi.disa.mil/brochure.html
2. U.S. Air Force, Corporate Information Division, "Data Strategy Paper," April 24, 1998.
3. Office of the Assistant Secretary of Defense Health Affairs memorandum, "Data Standardization Contract Language for Commercial-Off-the-Shelf Solutions," December 15, 1997, http://hirs.brooks.af.mil/mhss/cots.html
4. DoD 8320.1-M-1, Data Standardization Procedures, April 1998, http://www-datadmn.itsi.disa.mil.

Notes

1. The data administration organization consists of a DoD Data Administrator, Functional Data Administrators (at the secretarial level and often the principal staff assistants), and component data administrators (for separate agencies and military services). This is supplemented by a central organization providing services, such as central policy development, maintenance of the data models, and maintenance of data element standards contained in data repositories.
2. Other DoD data administration policies and procedures include: DoD Directive 8320.1, DoD Data Administration and DoD 8320.1-M, Data Administration Procedures.
3. Shared Data Engineering (SHADE). SHADE can be accessed through http://dii-sw.ncr.disa.mil.
4. Static data values include such things as country names, state names and abbreviations (codes) (i.e., Virginia, VA, Alabama, AL, etc.), postal zip codes, security classification codes, and the like.
5. Defense Information Infrastructure (DII) Common Operating Environment (COE).

Y2K Information from the Office of the Assistant Secretary of Defense

The Office of the Assistant Secretary of Defense (OASD) command, control, communications, and intelligence (C3I) Year 2000 team has established a Department of Defense-wide (DoD) listserver for the purpose of broadly disseminate year 2000 (Y2K) information. You are formally invited to add your e-mail address to this list. Throughout the next year, OASD (C3I) will use this server to distribute unclassified news highlights, technical developments, published policies, briefings, and DoD upcoming events. All information will be of a technical nature, and will be relevant to solving the Y2K problem.

How it Works

Simply follow the three steps to be added to the distribution.

- Send a message to: listserv@list servy2k.c3i.osd.mil
- Leave the subject line blank
- In the message body type: SUB y2k Your Name (for example: John Smith, not your e-mail address)

Once you have signed up, you will receive monthly news-mails published by the OASD (C3I) Y2K technical team. If at any time you no longer wish to be a member of the list, simply:

- Send a message to: listserv@list servy2k.c3i.osd.mil
- Leave subject line blank
- In the text section type: SIGnoff list

We look forward to providing you timely information via our listserver and serving the DoD community in this way. If you have any questions about the use of the listserver, please contact me, 1st Lt. George Hellstern at 703-602-0980 ext. 130 or Yogesh Patel at 703-602-0922 ext. 151, yogesh.patel@osd.pentagon.mil.

1st Lt. George Hellstern, USAF
OASD (C3I), Year 2000 Technical Services
george.hellstern@osd.pentagon.mil

Call for Articles

If your experience or research has produced information that could be useful to others, CROSSTALK will get the word out. We welcome articles on all software-related topics, but are especially interested in several high-interest areas. Drawing from our reader survey data, we will highlight your most requested article topics as themes for future CROSSTALK issues. In future issues, we will place a special, yet nonexclusive, focus on:

Handling Troubled Software Projects
December 1999
Article Submission Deadline: Aug. 2, 1999

Risk Management
January 2000
Article Submission Deadline: Sept. 3, 1999

Education and Training
February 2000
Article Submission Deadline: Oct. 1, 1999

We will accept article submissions on all software-related topics at any time; issues will not focus exclusively on the featured theme.

Please follow the *Guidelines for* CROSSTALK *Authors*, available on the Internet at http://www.stsc.hill.af.mil.

Ogden ALC/TISE
ATTN: Heather Winward
CROSSTALK Associate Editor/Features
7278 4th St.
Hill AFB, Utah 84056-5205

Or e-mail articles to features@stsc1.hill.af.mil. For more information, call 801-775-5555 DSN 775-5555.

Confusing Process and Product: Why the Quality is not There Yet

David A. Cook
Software Technology Support Center

For years now, the Department of Defense (DoD) and commercial software development organizations have embraced the Software Engineering Institute (SEI) Capability Maturity Model (CMM). In addition, there are many organizations that are rushing to meet the requirements of International Organizations for Standardization (ISO) 9000 and 9001. Unfortunately, organizations that meet CMM or ISO requirements are not necessarily producing quality software. This article discusses some impediments to software quality that remain in spite of CMM or ISO certification.

Quality Defined

Quality is a difficult thing to formally define. If you consider the strict definition, ISO 9001 suggests that it is "meeting requirements[1]." This is important, but not sufficient. In my experience, software that meets requirements is inadequate. Most software developers will quickly point out that many requirements are implied or implicit, often unstated, and frequently not addressed until implementation occurs. In addition, most end-users are concerned with reliability and robustness. Reliable software does what it is supposed to do, and does not do things it is not supposed to do. Robust software not only is reliable but also works dependably when confronted with unexpected or unanticipated conditions. Because software systems today are so large and complex, and are often expected to work under severe conditions where failure could mean loss of human life, these systems need to be robust.

The CMM and ISO

Regardless of how you feel about quality, reliability, or robustness as criteria, we all agree that most, if not all, software needs to have improved quality. Before we can improve the quality, however, we need to determine how the software is built. And for that, we need to define how we built the software. One of the best efforts in recent years to improve how we engineer software has been the SEI CMM [1]. The CMM still is not universally respected by all practitioners but it unquestionably alerts an organization to the practices that must exist for good, reliable software engineering to be performed.

Many DoD organizations have achieved CMM Level 3 (the "defined" level). The definition of this level is that the software processes for both management and engineering activities is documented, standardized, and integrated into a standard software process for the organization. In effect, Level 3 of the CMM removes the "superprogrammer" as the main reason for a company achieving good software. Watts Humphrey said, "There is a common view that a few first-class artists can do far better work than the typical software team. … If this were true, one would expect that those organizations that have the best people would not suffer from the common problems of software quality. … Experience, however, shows that this is not the case" [2]. CMM Level 3 requires an organizational process that tries to overcome the "superprogrammer" mindset and focuses on sound software engineering principles for all developers.

Note, however, that organizational processes are usually insufficient for truly great software — software is still developed by individuals. Personal processes are still necessary — which is why several organizations have experienced great success in improving quality by using the Personal Software Process (PSP). PSP requires the equivalent of CMM for individual programmers — a process that addresses quality on an individual, not organizational, level [3]. PSP, when used in conjunction with the CMM, provides personal processes that complement the organizational processes, providing a better chance of quality. As another weapon in the fight against poor quality, some organizations look to ISO 9001. After all, it's very title implies that following it will produce a quality system. Unfortunately, most people confuse a quality system — which is what ISO 9001 is concerned with — a quality product. Quality systems are necessary to ensure the development of a quality product but they are not sufficient [4]. To make matters worse, ISO 9001 is not even sufficient for a complete quality system.

W. F. Fightmaster, vice president of quality for Square D (part of France's Groupe Schneider) said, "There are some people who believe that once you have ISO you have a quality system. That just isn't so. It is less than one-seventh of the system [5]." Still, it is a fact that a quality system is required if you want the end product to have quality. In mathematical terms, a quality system is necessary but not sufficient for a quality product. One customer warned me to be wary of ISO 9001 — that it was possible to design a quality system that produced concrete life jackets.

Why We Do Not Have Quality Yet

The point of this article is that we have tools that work. The CMM improves the organizational process, PSP improves the individual process, and ISO 9001 provides a quality system. The question remains: Why are we not seeing great increases in quality?

Based on several experiences, I can now point out three problem areas where most organizations fail. In my opinion, all

three "truths" are obvious. Perhaps that is why I need to state them, because obvious truths are sometimes the hardest to see and understand.

- *We are not using common sense.*
- *We have one process we publicize and another process we use.*
- *Good practices cannot overcome really poor management.*

Tailor the Process to Your Needs

Truth No. 1: We are not using common sense. The Capability Maturity Model is a process, not a product. Achieving CMM Level 3 is not the end, quality software is the end. I recently worked with an organization that wanted to organize their software engineering process group (SEPG) to help it achieve and coordinate development of its software process. It located another organization that had recently organized their own SEPG and copied the documentation almost directly. One problem — the organization they borrowed from had nearly 300 developers, plus several levels of management, while its own organization consisted of 17 people. Imagine, a 17-person shop following guidelines set up for a 300-person organization. A skimming of their SEPG documentation convinced me that it would have to spend more than 50 percent of their work time in SEPG-related meetings. Yet, the organization managed to achieve their CMM Level 3 — in spite of the fact that they could not produce software within their own process.

The point is not that common sense does not exist, it is that we forget the difference between the product (quality software) and the process (following the CMM). The CMM should be a "living process," in that frequent reviews lead an organization to self-improvement. This is the purpose and intent of the CMM, yet most organizations I have worked with treat the CMM-related documentation as a standard. Many of the individual developers treat the CMM with the same loathing that we used to regard Military Standard 2167A. MilStd2167A now is regarded for its perceived imposition of the waterfall development model, inflexibility related to object-oriented design, excessive documentation, no guidance on management indicators, and the need to incorporate new development techniques such as reuse and re-engineering [6]. Yet, some organizations have set into place processes that are equally inflexible in similar ways. We cannot afford to put processes in place that do not work.

As a further comment on the lack of common sense, a recent SEI monograph [7] discussed some problems with a government project. Integrated process teams were not integrated — there was a "government" side and a "contractor" side. This monograph is well-worth reading — it points out where common sense was lacking.

Use a Process that Works for You

Truth No. 2: We have one process we publicize and another process we use. It is my firm belief that most organizations I have worked with produce good software because most of the low-level developers have internalized the work-arounds in the system. The process does not work and is not modified. Yet, the developers have found ways to make the system work in spite of the documentation and process. This is a yardstick that I use when I consult with an organization: If the developers are not really following the process, then the process does not work. This is not saying that developers will automatically follow a good process; I think that good software developers have some type of genetic defect that makes them want to buck the system most of the time. But software developers can innately tell when the process will or will not work, and will follow a process once they are convinced that it is beneficial to them. One organization I worked with had developers that fought an organized review process. They fought until they saw the benefits in terms of rework and maintenance. If all developers are ignoring a process, then the "public process" is for show and the "hidden process" is the one that works.

Management Needs Common Sense, too

Truth No. 3: Good practices cannot overcome really poor management. When I was little, my heroes were Superman and Batman. Now, my hero is Dilbert (actually his creator, Scott Adams). His "Pointy-Haired Boss" [8] seems to typify what is wrong with software engineering. Every software development organization I visit has Dilbert cartoons posted. Why? Because the problems seem to hit home. Gerald Weinberg, in his book, *The Psychology of Computer Programming* [9] says, "bad supervision and leadership is more common than we would like to imagine." A recent customer I worked with had totally separated the developers from the analysts, and the analysts from the functional domain experts. The reason, supposedly, was to "improve communication by providing single-point interfaces." The real reason, of course, was a turf battle. This turf battle, where several managers were unable to allow free communication between co-workers under their control, resulted in software that could neither be verified nor validated. Managers need training in current practices and techniques, and they need to have a buy-in for the ISO and CMM. If management does not understand what is expected, they cannot be blamed for not following the process. Here is a sad fact — managers who have not really changed their processes since the '60s (and still think that the "waterfall model" is just a new-fangled, passing fancy) will never be able to creatively lead a team that produces quality software.

Often, the case is not even that managers are old-fashioned. Frequently they know nothing. With great regularity, we still have cases where medium- to large-scale software acquisition occurs by managers who do not understand the basic fundamentals about requirements and contracting. Asking "What are my requirements?" after half a million dollars has been spent on unusable software might stimulate the national economy, but only causes frustration on the part of the poor users trying to use a system that does not meet any of their needs. In short, developing and procuring software requires the expertise of people who have training and experience in software development and acquisition. Managers who ignore this advice and attempt to do it on their own end up with useless systems. Frederick

Brooks must have had these managers in mind when he made the observation, "Plan to throw one away: you will, anyway [10]." Alan Davis says that good management stifles motivation and erases good work they have accomplished [11].

Summary

So what is the result? Throw CMM and ISO to the winds? No. Proponents of CMM and ISO need to dig in harder. ISO 9001 and the CMM might not be the ultimate tools but they are the best tools we have. What is needed is a healthy dose of common sense. Quality software requires a process, and the process must be different for each particular organization. In addition, the process must be self-modifying and dynamic, again to meet the specific needs of each product and organization. Remember that quality software is the end, and that the process is the means. If we keep our eyes on the target and modify the process to allow us to reach it, quality software can be produced.

About the Author

David Cook is a principal member of the technical staff, Charles Stark Draper Laboratory, under contract to the STSC. He has more than 25 years experience in software development and has lectured and published articles on software engineering, requirements engineering, Ada, and simulation. He has been an associate professor of computer science at the U.S. Air Force Academy, deputy department head of the software engineering department at the Air Force Institute of Technology, and chairman of the Ada Software Engineering Education and Training Team. He has a doctorate degree in computer science from Texas A&M University and is an SEI-authorized PSP instructor.

Software Technology Support Center
7278 Fourth Street
Hill AFB, Utah 84056
Voice: 801-775-3055
Fax: 801-777-8069
E-mail: cookd@software.hill.af.mil

References

1. Paulk, M. et. al. "Capability Maturity Model for Software," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa. 1993.
2. Humphrey, W., *Managing the Software Process*, Addison Wesley, 1990.
3. Humphrey, W., *A Discipline for Software Engineering*, Addison Wesley, 1995.
4. Radice, R. "ISO 9001 Interpreted for Software Organizations," Paradoxicon Publishing, 1995.
5. Henkoff, R., "The Hot New Seal of Quality," *Fortune magazine*, June 28, 1993, pp. 116-120.
6. Newbery, G. "Changes from DOD-STD-2167A to MIL-STD-498," *CROSSTALK* The Journal of Defense Software Engineering, April 1995.
7. Carney, D. "Case Study: Signigicant Schedule Delays in a Complex NDI-Based System," SEI Monographs on the Use of Commercial Software in Government Systems, Software Engineering Institute, Carnegie Mellon, Pittsburgh, Pa. 1998.
8. Adams, S. *Seven Years of Highly-Defective People*, Andrews McMeel Publishing, 1997.
9. Weinberg, G., *The Psychology of Computer Programming*, Silver Anniversary Edition, Dorset House, 1998.
10. Brooks, Frederick P. Jr., *The Mythical Man-Month*, Addison Wesley, 1995.
11. Davis, Alan M., *201 Principles of Software Development*, McGraw-Hill, 1995.

Note

1. ISO 9001, Quality system — Model for quality assurance in design, development, production, installation, and servicing.

Coming Events

Software Engineering Symposium '99

Dates: Aug. 30-Sept. 2
Location: David L. Lawrence Convention Center, Pittsburgh, Pa.
Early-bird registration: July 28
Theme: Improving the State of Software Engineering — Principles, Practices, and Projections. The program will have plenary sessions, tutorials, panels, exhibits, and birds-of-a-feather sessions. The symposium will be held in conjunction with the conference on Software Technology and Engineering Practice.
Voice: 412-268-3007
Fax: 412-268-5556
E-mail: symposium@sei.cmu.edu
Internet: http://www.sei.cmu.edu/products/events/symp/ or http://www.iwcase.org/step99/

ESEC/FSE '99 Joint 7th European Software Engineering Conference (ESEC) and 7th ACM SIGSOFT International Symposium on the Foundations of Software Engineering (FSE-7)

Dates: Sept. 6-10, 1999
Location: Toulouse, France
Topic: ESEC/FSE '99 will bring together researchers and practitioners of modern software engineering techniques to exchange new research results and reports related to traditional and emerging fields of software engineering. ESEC/FSE '99 also will include a program of tutorials and workshops on current topics in software engineering.
Sponsors: ACM, SIGSOFT, CEPIS, SUPAERO, and ONERA
Internet: http://www.iam.unibe.ch/~esec99/

2nd USENIX Conference on Domain Specific Languages (DSL '99)

Dates: Oct. 3-6, 1999
Location: Austin, Texas
Sponsors: USENIX, the Advanced Computing Systems Association, in cooperation with ACM SIGPLAN and SIGSOFT
Internet: http://www.usenix.org/events/dsl99/

The Second International Conference on The Unified Modeling Language — UML '99

Dates: Oct. 28-30, 1999
Location: Fort Collins, Colo.
Objective: UML '99 will bring together researchers in academia and industry who are developing processes, methods, techniques, and semantic foundations for the UML. The conference will provide a forum for discussing and evaluating promising approaches that will enhance the application of UML.
Sponsors: IEEE Computer Society Technical Committee on Complexity in Computing in cooperation with ACM SIGSOFT.
Contact: Robert France, conference chairman
Voice: 970-491-6356
Fax: 970-491-2466
E-mail: France@cs.colostate.edu

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 theme will focus on 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

CROSSTALK

SPONSOR Lt. Col. Joe Jarzombek

PUBLISHER Reuel S. Alder

ASSOCIATE PUBLISHER Tracy Stauder

EDITOR IN CHIEF Brent Baxter

MANAGING EDITOR Kathy Gurchiek

ASSOCIATE EDITOR/LAYOUT Heather L. 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/crosstalk.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*.

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 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: This 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.

Farewell from Lorin May

Farewell, dear readers, this is the last time I get to insult your collective intelligence. On April 1, I officially stopped working for *CROSSTALK* and became a bona fide contractor techie.

Stop laughing — I'm one of you now. Not to brag, but I frequently impress co-workers with my knowledge of C++, B++, gooeys (and how to avoid them), and how to IF-THEN-ELSE every oriented object in sight. In fact, judging from the quizzical looks, I think I'm often talking over their heads.

Perhaps you are thinking, "You, the guy who thinks MacGyver is a real person, and that an algorithm is a funky dance step, are allowed to be in the same room as actual code?"

I'm sick of hearing that question. Besides, I'm working in the requirements phase, and any *CROSSTALK* reader knows that if you are going to make mistakes, requirements is the phase in which to do it.

It's nice to be able to look back on my three years with *CROSSTALK* and think, "I can be proud that I produced 35 issues of the military's premiere software journal without making a single error that was conclusively linked to a software-glitch-induced international crisis." I even have the acquittal papers to prove it.

And *CROSSTALK* was a great springboard for some of my software-related accomplishments, including:

- Developing the first comprehensive, commercially viable Y2K solution endorsed by the Gouda Cheese Makers of America (GCMA).

- Championing the consolidation of various leading software standards, frameworks, and models into one simple, user-friendly framework called MIL-STD-ISO-498-CMMI-9000-OMG-SEI-IEEE-BVD-DII-COE-CIA-CORBA-DVD-DOA-RSVP-ASAP-2167A version 2.0 (draft), also known in professional circles simply as "MSI4C9OSIBDCCCDDRA2V2D."

- Teaming with Al Gore to develop the LINUX operating system.

- Designing and coding my first software program, a modest little system called SoftComplete 1.0. At the touch of a button it automatically converts written requirements into bug-free code, writes complete documentation, and configures and installs the system to multiple-user environments in distributed locations, all in less than five minutes. Or at least it will in the next release. (System requirements: Windows 3.x, Dual 300 GigaHertz processors, 512 Terabytes of RAM, 4,194,304 Trilobites of disk space, 5 1/4" floppy drive.)

So farewell, it's been fun. If you want to chat, or wish me good riddance, drop me a line at lorinmay@hotmail.com.

— Lorin May, former associate editor

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