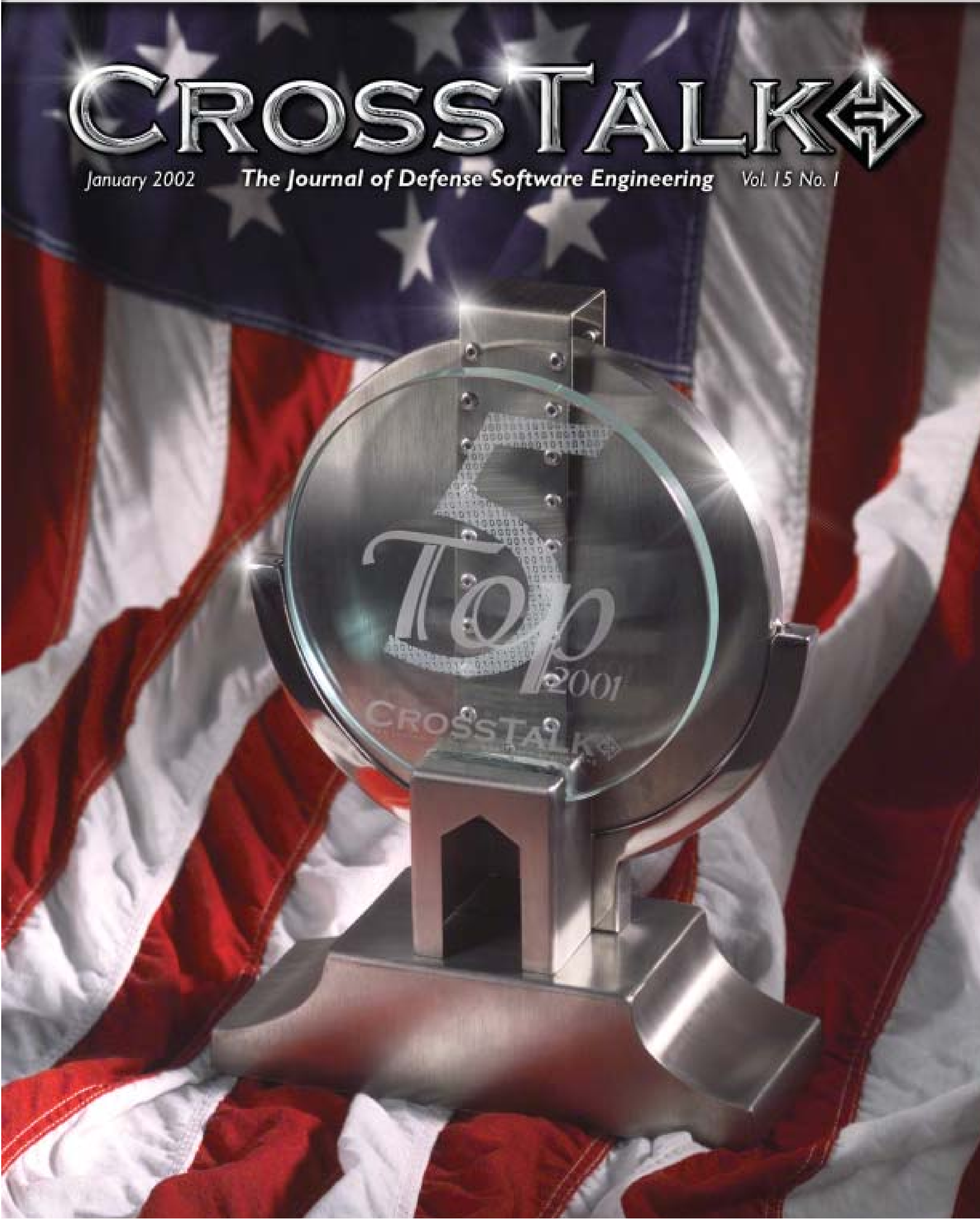
SPECIAL SOFTWARE QUALITY AWARDS ISSUE

CrossTalk

January 2002 The Journal of Defense Software Engineering Vol. 15 No. 1



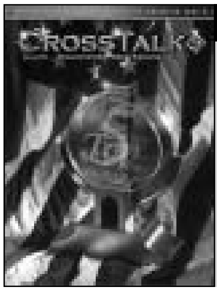
Top Five Quality Software Projects

Best Practices

Software Engineering Technology

Open Forum

Departments



On The Cover

Kent Bingham, Digital Illustration and Design, is a self-taught graphic artist/designer who freelances print and Web design projects.

CrossTalk

Sponsor	*Lt. Col. Glenn A. Palmer*
Publisher	*Tracy Stauder*
Associate Publisher	*Elizabeth Starrett*
Managing Editor	*Pam Bowers*
Article Coordinator	*Nicole Kentta*
Creative Services Coordinator	*Janna Kay Jensen*
Phone	(801) 586-0095
Fax	(801) 777-5633
E-mail	crosstalk.staff@hill.af.mil
CrossTalk Online	www.stsc.hill.af.mil/crosstalk/crostalk.html
CRSIP Online	www.crsip.hill.af.mil

Subscriptions: Send correspondence concerning subscriptions and changes of address to the following address. You may e-mail or use the form on p. 33.

Ogden ALC/TISE
7278 Fourth St.
Hill AFB, UT 84056-5205

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 Author Guidelines, available at www.stsc.hill.af.mil/crosstalk/xtlkguid.pdf. CrossTalk does 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: www.stsc.hill.af.mil
Call (801) 777-7026, e-mail: randy.schreifels@hill.af.mil

Back Issues Available: The STSC sometimes has extra copies of back issues of CrossTalk available free of charge.

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 to improve the quality of their software products, efficiency in producing them, and their ability to accurately predict the cost and schedule of their delivery.

Now More Than Ever, Software Is the Heart of Our Weapons Systems

On June 15, 2001, I took over the organizational management responsibility for the Office of the Secretary of Defense Software Intensive Systems Directorate and Steering Group from Dr. Delores Etter. I am delighted to have this opportunity to strengthen my organization's capabilities and to increase our involvement with software engineering in the acquisition, deployment, and sustainment of the critical systems supporting our warfighters. Now, more than ever, software is the heart and soul of our weapons systems. We must focus our full range of resources on meeting the very difficult challenges required to gain the vast benefits afforded us by software intensive systems. Along with my new Deputy Director for Software Intensive Systems Joe Jarzombek, I look forward to working with all of the defense software collaborators, including the Software Technology Support Center, as we continue to improve the quality, effectiveness, and responsiveness of defense acquisitions.

It seems fitting that at the beginning of a new year, one that finds all of America reevaluating priorities and facing the world with perceptions considerably changed from just a few months ago, CrossTalk has the privilege of recognizing the Top 5 U.S. Government Award-Winning Quality Software Projects. These five projects, representing the best of our government's software capabilities, were selected from 87 nominees by a panel of highly qualified and experienced judges. The projects were evaluated on customer satisfaction as well as quality and performance.

These projects, each described in detail in articles this month, represent a variety of software challenges that benefit both government and civilian sectors. The "Standard Terminal Automation Replacement System" is a joint program between the military and the Federal Aviation Administration that will bring substantial benefits to the flying public as well as enhance safety in both civilian and military airspace. The "Higher Authority Communications/Rapid Message Processing Element" product supports the readiness and control of our nation's nuclear missile capability. The "F/A-18 Advanced Weapons Lab System Configuration Sets" provide the operating configurations and logistics support documentation for more than 10 million words of avionics, weapons, and flight control software in an environment that includes more than 40 different processors. The entire country is impacted by the "Data Capture System 2000" program that automatically processed more than 150 million multi-page census forms in 170 days. Tactical effectiveness is the goal of the "Force XXI Battle Command Brigade and Below" project that represents a revolutionary change in warfare for the Army.

Beyond congratulating successful, award-winning projects, this issue also supports beginning and ongoing projects. Dr. Barry Boehm and his Center for Software Engineering researchers describe some effective tools for implementing evolutionary acquisition in their article *Using the Spiral Model and MBASE to Generate New Acquisition Process Models: SAIV, CAIV, and SCQAIV*. It has been an honor to share the work of this outstanding group with CrossTalk readers. *Project Recovery … It Can Be Done* gives Walt Lipke's sage advice on what to do when software projects go awry. Finally, John Michel's update on the impact of the Clinger-Cohen Information Technology Management Reform Act, *CIO Update: The Expanding Responsibilities*, provides a context for its application to future Department of Defense systems.

This looks to be a challenging year for America, particularly for its military and government workforce. Our air traffic system is adapting to new realities in order to provide our economy with safe and efficient transportation. Our warfighters are deployed and doing the jobs they have been trained to do, so that all of us can recover our sense of freedom, safety, and security. They are using equipment and systems that we have acquired, developed, tested, deployed, and sustained. They are the best equipped fighting force the world has ever known, and our job is to make sure that they stay that way. Through new technology, agile minds, and hard work, we must provide them the capability to rapidly respond to new threats here and abroad. The projects honored here are doing just that. Congratulations to all.

Dr. Nancy Spruill
Director, Acquisition Resources and Analysis
Office of the Under Secretary of Defense (AT&L)

Greatness in Software Development Abounds in the U.S. Government

Pamela Bowers
CrossTalk

CrossTalk is proud to announce the winners of the 2001 Top 5 U. S.Government Quality Software Projects. These top five projects were selected from 87 nominations in this first event. They demonstrate how competent software project teams go about building successful products. In the following pages, we present a brief article on each winning project, along with biographies of the judges, and brief summaries of the top 11 finalists.

U.S. Government's Award-Winning Quality Software Projects

The results are in, and it is clear that the government is building many successful software packages that are top-notch examples in project management and quality control. Listed below in alphabetical order are the winners in CrossTalk's first Top 5 U.S. Government Quality Software Projects contest. We congratulate them and hope you enjoy reading more about their winning projects in the following pages.

- **ATB-230 STARS Product Team**
  Project Name: Standard Terminal Automation Replacement System
  Customer Organization: National Air Traffic Controllers Association, and Professional Airways Systems Specialists
  Location: Washington, D.C.

- **Detachment 1, Ogden ALC**
  Project Name: Higher Authority Communications/Rapid Message Processing Element
  Customer Organization: Headquarters, Air Force Space Command
  Location: Offutt AFB, Neb.

- **F/A-18 Advanced Weapons Lab**
  Project Name: System Configuration Set 15C
  Customer Organization: Program Manager, Air-265
  Location: China Lake, Calif.

- **Lockheed Martin Mission Systems**
  Project Name: Data Capture System 2000
  Customer Organization: Bureau of the Census
  Location: Bowie, Md.

- **TRW Tactical Systems Division**
  Project Name: Force XXI Battle Command Brigade and Below
  Customer Organization: U.S. Army CECOM

Great software projects are a combination of excellence in project management, estimation, measurement, quality control, and change management. This type of performance is exhibited by the 2001 U.S. Government's Top 5 Quality Software Projects announced here in CrossTalk. The winners and finalists alike in this first CrossTalk presentation are "existence proofs that large and complex software projects can be finished on time, within budget, meet with favorable user reactions, and have few remaining defects after delivery," says Capers Jones, chief scientist of Artemis Management Systems and director of Software Productivity Research Inc., Burlington, Mass., and a Top 5 judge.

Tough Choices

There were 87 nominations received in CrossTalk's search for the top five government software projects. It was extremely difficult to narrow this list. Each nomination was scored by at least three professionals from the Software Technology Support Center (STSC) at Hill Air Force Base. The STSC consists of software engineers and consultants that have worked in both software engineering process groups and in Capability Maturity Model® Level 5 software development projects, and are recognized as expert authors and speakers on software.

The STSC reviewers judged projects on customer value, performance, technical value, and per their own discretion (see page 18 for complete scoring parameters). From this scoring, the top one-third of nominations was selected. The project customers were contacted to ensure product satisfaction and their high regard for the project teams. The remaining top nominations, along with their customer questionnaires and additional information, were sent to seven judges, scored from one to 100, and ranked from one to 16. Judges were free to use their own scoring criteria, but were required to provide justifications for all projects ranked one through five. In most cases, judges used a similar ranking system as the STSC reviewers.

These preliminary scores were used to select the top eight projects for final scoring. Judges then reviewed the other judges' justifications for these eight projects to decide the final top five projects on Sept. 5, 2001. These five projects are all ranked equally as winners (See accompanying sidebar). A more detailed look at each project follows in this special Top 5 section.

CrossTalk was very fortunate to have some of the best and well-known software professionals volunteer as judges for the 2001 U.S. Government's Top 5 Quality Software Projects. They gave considerable time to reviewing entries and writing summary decisions. The CrossTalk staff is proud to give you a brief biography of each judge on page 5. Without them, the contest would not have been possible. We also thank all the STSC reviewers for their time and effort, and everyone who provided nominations.

Each Top 5 winner will be presented with an engraved award like the one featured on this CrossTalk cover during the 2002 Software Technology Conference (STC) "Forging the Future of Defense Through Technology" in Salt Lake City April 28-May 2. The award ceremony will be held in front of an estimated 2,000 STC attendees and the leaders of each of the Department of Defense services on Tuesday, April 30 during the co-sponsored panel discussion from 8:00-9:30 a.m.. A Top 5 panel discussion will also be held on Wednesday, May 1.

It is our hope to continue to award the top government software projects annually. The U.S. government is doing a great job of producing software to support the entire country in many, many ways. CrossTalk's Top 5 U.S. Government Quality Software Projects award recognizes that greatness does abound in U.S. government software project teams.◆

Top 5 Quality Software Projects Judges' Biographies

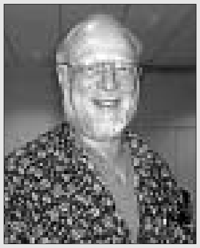
David A. Cook, Ph. D., is a principal engineering consultant at Shim Enterprise Inc. He is currently assigned as a software-engineering consultant to the Software Technology Support Center, Hill AFB, Utah. He was formerly an associate professor of computer science at the U. S. Air Force Academy, and also a former deputy department head of the Distance Education Department at the Air Force Institute of Technology. Dr. Cook was a member of the Air Force Ada 9x Government Advisory Group, and has published numerous articles on software process improvement, software engineering, object-oriented software development, and requirements engineering. He has a bachelor's degree in computer science from the University of Central Florida, a master's degree in teleprocessing from the University of Southern Mississippi, and a doctorate in computer science from Texas A&M University. Dr. Cook's e-mail address is <david.cook@hill.af.mil>.

Carol Dekkers is president of Quality Plus Technologies Inc., a management consulting firm specializing in helping Department of Defense and private organizations succeed with function points, make wise investments in software measurement, and achieve bottom-line improvements through process improvement. Dekkers is vice-chair of the Project Management Institute Metrics Specific Interest Group, a past president of the International Function Point Users Group, and an International Organization for Standardization project editor on the Functional Size Measurement project. She was named one of the 21 New Faces of Quality for the 21st century by the American Society for Quality. She is a professional engineer, certified function point specialist, and a certified management consultant. Dekkers e-mail address is <dekkers@qualityplustech.com>.

Jack Ferguson, Ph.D., is deputy Acquisition Resources and Analysis director for Software Intensive Systems. Dr. Ferguson is responsible for improving the research, development, and acquisition of software intensive systems in the Department of Defense (DoD). Formerly Dr. Ferguson was a senior member of the Technical Staff at the Carnegie Mellon University Software Engineering Institute (SEI). He led the project to develop the Software Acquisition Capability Maturity Model® and was program manager for Capability Maturity Model®-Integrated^{SM}. Dr. Ferguson also spent 26 years in the U.S. Air Force. He has a doctorate in aerospace engineering from the University of Texas at Austin. He won the Air Force Research and Development Award for his work on attitude control of GPS spacecraft and is listed in Jane's Who's Who in Aerospace. Dr. Ferguson's e-mail address is <fergusj@acq.osd.mil>.

Watts S. Humphrey is a fellow at the Software Engineering Institute (SEI) of Carnegie Mellon University, which he joined in 1986. At the SEI, he established the Process Program, led initial development of the Capability Maturity Model®, introduced the concepts of Software Process Assessment and Software Capability Evaluation, and most recently, the Personal Software Process and Team Software Process. Prior to joining the SEI, he spent 27 years with IBM in various technical executive positions. He has a master's degree in physics from the Illinois Institute of Technology and in business administration from the University of Chicago. He is the 1993 recipient of the American Institute of Aeronautics and Astronautics Software Engineering Award. His most recent books include *Managing the Software Process*, *A Discipline for Software Engineering*, *Managing Technical People*, and *Introduction to the Personal Software Process*. Humphrey's e-mail address is <watts@sei.cmu.edu>.

Capers Jones is chief scientist emeritus of both Artemis Management Systems and Software Productivity Research Inc., Burlington, Mass. Jones is an international consultant on software management topics, a speaker, a seminar leader, and author. He is also well known for his company's research programs into the following critical software issues: Software Quality: Survey of the State of the Art; Software Process Improvement: Survey of the State of the Art; Software Project Management: Survey of the State of the Art. Formerly, Jones was assistant director of programming technology at the ITT Programming Technology Center in Stratford, Conn. Prior to that he was at IBM for 12 years. He received the IBM General Product Division's outstanding contribution award for his work in software quality and productivity improvement methods. Jones' Web site is <www.spr.com>. His e-mail address is <cjones@spr.com>.



Gerald "Jerry" M. Weinberg, Ph. D., is a principal in the consulting and training firm of Weinberg & Weinberg. For more than 45 years, Dr. Weinberg has worked on transforming software organizations. He is author or co-author of more than 40 books, including *The Psychology of Computer Programming* and *Introduction to General Systems Thinking*. Many of his books cover all phases of the software life cycle, including *Exploring Requirements*; *Rethinking Systems Analysis and Design*; *The Roundtable on Project Management*. His books on leadership include *Becoming a Technical Leader*, *The Secrets of Consulting*, *Quality Software Management* four-volume series, and more. He is also known for his workshops for software leaders, including "Problem Solving Leadership (PSL)," "The Congruent Organizational Change-Shop," and more. Dr. Weinberg's Web site is <www.geraldmweinberg.com>. His e-mail address is <hardpretzel@earthlink.net>.

Brenda Zettervall is a computer scientist at the Naval Surface Warfare Center, Dam Neck. She has more than 25 years of experience in the field of computer systems and software engineering for complex, real-time command and control systems used both for deployment on U.S. Navy surface ships and for simulated land-based integration. Currently Zettervall is deputy director for the Naval Collaborative Engineering Environment, the office of the Chief Engineer for the Navy, for Research, Development, and Acquisition. In addition, Zettervall provides naval coordination for software related activities for the Department of Defense Software Intensive Systems Steering Group. She was awarded the Navy's Meritorious Civilian Service award for her work with the Office of the Undersecretary of Defense. Zettervall has a bachelor's degree in mathematics from Radford University. Her e-mail address is <zettervallbt@navsea.navy.mil>.

The FAA Replaces an Aging Government and Commercial Air Traffic Control System

Pamela Bowers
CrossTalk

The Standard Terminal Automation Replacement System (STARS) represents a dramatic change in performance of a Federal Aviation Administration system. STARS is a state-of-the-art air traffic control system that will ease the burden on air traffic controllers making the system easier to use and more convenient. It will save a tremendous amount on logistics and support costs, and being digital, additional interfaces can be added as needed.

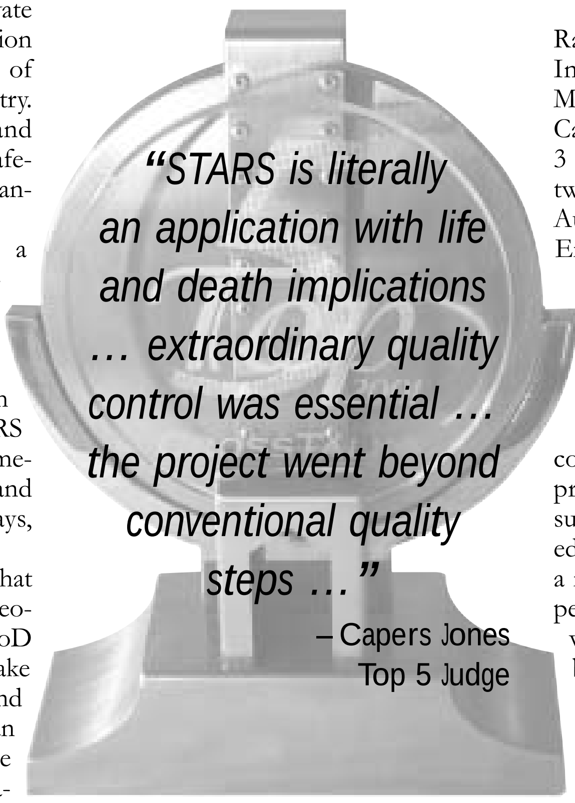
The Standard Terminal Automation Replacement System (STARS) is a state-of-the-art air traffic control system that provides approach and departure control for commercial, military, and private aircraft. It replaces aging Federal Aviation Administration (FAA) and Department of Defense (DoD) systems across the country. STARS provides increased arrival and departure efficiency and incorporates safety features, including quadruple redundancy and automatic emergency back up.

The delivered product consists of a site-dependent number of terminal control workstations used to control air traffic, and to monitor and control workstations. These are used to monitor and configure the system, perform maintenance, etc. Although the STARS system is "standard," it is entirely parameter-adaptable to accommodate site and operational differences (i.e., runways, radars, etc).

STARS is a safety-critical system that will affect the daily lives of millions of people flying in and out of 331 FAA and DoD sites. The success of STARS will make traveling the nation's airways safer and more convenient. This is due in part to an innovative design decision to minimize air traffic controllers' training, familiarization, and certification by retaining the look and feel of the previous air traffic control system. STARS development provided for the air traffic controllers and Air Force system specialists to be involved from the start in an extensive series of product and development assessment demonstrations, early user involvement events, and a variety of Software Trouble Report/Program Trouble Report (STR/PTR) and Computer Human Interface (CHI) working groups. This involvement was necessary to expose the air traffic control community to the system's features and to incorporate their feedback into all new development iterations.

COTS Was First Choice

STARS is based on open-system architecture using commercial off-the-shelf (COTS) technology that provides future extensibility. To reduce complexity of support sites a single, uniform hardware and software architecture was used. This minimizes life-cycle costs for training, maintenance, and sparing. Development, site, and support facility processors use Sun Microsystems with Sun Solaris operating systems using the C language in a Unix development environment with ClearCase configuration management.



> "STARS is literally an application with life and death implications … extraordinary quality control was essential … the project went beyond conventional quality steps …"
>
> – Capers Jones
> Top 5 Judge

To improve display-processing speed associated with commercial display drivers, upgraded TechSource boards were installed in both the Emergency Service Level (ESL) and Full Service Level (FSL) processors. This allows both systems to run concurrently, updating their internal displays. A software switch allows either of the displays to be quickly brought up on the monitor, improving refresh time for quick cut-overs.

The STARS development contractor, Raytheon's Command, Control, and Information Systems in Marlborough, Mass., is a Software Engineering Institute Capability Maturity Model® (CMM®) Level 3 organization. Raytheon's design employs two commercially available products: AutoTrac-Full Service and TracView-Emergency Service. Using separate non-developmental item (NDI) software for full and emergency service was Raytheon's solution to isolation of problems between the operational system and the backup system.

Raytheon based their source-line-of-code (SLOC) estimates on its historic product metrics database. Government support personnel independently estimated SLOC by decomposing requirements to a module level, then estimating the SLOC per module. Government estimates were validated by function point analysis, and both estimates were reconciled. Since that time, the actual SLOC developed and delivered for STARS has remained within 5 percent of the SLOC estimates.

As customer familiarization with the system progressed through much human interface prototyping, it was decided for safety reasons that the STARS software should be changed to have the look and feel of the current system. (Human factor studies show that under stress or fatigue controllers can revert to old learned habits, leading to mistakes.) Although this drastically reduced retraining and increased safety, it more than doubled the software development effort and required new SLOC estimates for succeeding phases.

The Government's independent cost estimate was developed using three software models (COCOMO II, SLIM, and SEER-SEM). Each model was fine tuned with Raytheon's historic data to reflect realistic productivity numbers and, where

® *Capability Maturity Model and CMM are registered in the U.S. Patent and Trademark Office.*

Top Photo: *STARS ATB-231 at Northeast Operational Support Facility (NEOSF), Nashua, N.H.* Bottom Left Photo: *STARS ATB-230 at Washington, D.C.* Bottom Right Photo: *STARS ATB-232 at FAA Technical Center, Atlantic City, N.J.*

possible, the model was further adjusted to reflect new development to pre-existent code. The results of the three models were evaluated and compared to produce the most realistic schedules. STARS development has tracked closely to those schedules. Most STARS schedule slippage occurred in post-development, especially in Operational Test and Evaluation.

Quality Makes the Difference

What makes STARS different is the project's extraordinary commitment to achieving quality and customer satisfaction. Air traffic controllers have high pressure, high stress, and safety critical jobs that demand a system that always operates as expected. STARS has the same functionality as the existing systems. In addition it has higher performance, better reliability, more redundancy, and the capability to grow and be enhanced.

Capers Jones, a Top 5 judge said, "This project is important to both military and civilian air travel. It represented careful development practices and much better than average quality control."

STARS developed innovative ways to involve the air traffic controllers and Airways Facility system specialists on a rotating basis from the start in an extensive series of product and development assessment demonstrations, early user involvement events, and a variety of STR/PTR and CHI working groups. This involvement was necessary to expose the air traffic control community to the features of the system and to incorporate their feedback into each new development iteration.

"Thin Specs" were developed to capture CHI requirements at a level of detail significantly below the usual System/Subsystem Specification (SSS) level. Functional verification testing of the system was instituted in parallel with formal system acceptance tests in order to find and fix potential operational problems early. STARS was the first FAA program to implement new security requirements. It has become a model for other FAA projects for its innovative solutions.

Overall the results were impressive. "STARS is literally an application with life and death implications," says Jones. "Therefore extraordinary quality control was essential. The STARS project went beyond conventional quality steps and included some innovative methods for improving human factors and making the system easier to learn and use by air traffic controllers."

STARS ensures product, project, and process quality through application of recognized engineering practices, including CMM Level 3 for software development, ISO standards 9001/9003, and Six Sigma engineering practices for quality. Raytheon's approved quality system plan integrated the FAA's quality engineering procedure to ensure full compliance.

STARS software is subjected to rigorous inspection and test through all acquisition phases. Quality acceptance standards are also imposed on commercial product vendors. All replacement products and upgrades are thoroughly tested for backward and forward compatibility and interoperability with existing STARS products. A hierarchy of Raytheon and government change control boards performs baseline and requirements maintenance.

Measurement, including the cost performance index (CPI) and the project schedule performance index (SPI), are presented routinely at monthly program management reviews. Currently the CPI is 1.03 and the SPI is 0.98, which reflects the program is running within 3 percent of plan. Raytheon uses an earned value management system that fully integrates schedule, performance, and cost data. This data is made available at the end of each month to all personnel. The master integrated program schedule (MIPS) is also integrated into this process.

Within the STARS program, defects receive high visibility and tracking through a number of rigorously monitored means, including monthly program management reviews, biweekly presentations to the project manager, and weekly PTR working group meetings (PTRWG). The purpose of the PTRWG is to classify PTRs, assess PTR symptoms, review proposed resolutions to anomalies, and review analysis of root cause. Raytheon conducts root cause analysis of designated PTRs and provides recommendations for corrections to root causes. The government and Raytheon keep duplicate PTR databases with running totals of the number of PTRs, the time the PTR has been open, and SLOC per PTR closure. This data is used for statistical evaluation.

Success on All Fronts

STARS is the first large procurement program under the Reformed Acquisition Management system. STARS accomplished the total acquisition process in half the normal time.

"Innovations in development included aggressive use of cost, schedule and performance metrics, and the involvement of air traffic controllers throughout development," says Jack Ferguson, a Top 5 judge.

STARS has received very positive feedback and has encountered an unusually low number of problems from its three operational sites: Eglin AFB, Fla.; El Paso, Texas; and Syracuse, N.Y. Both the military and FAA air traffic control communities are eagerly awaiting implementation of full STARS at the remaining sites. To demonstrate the usability of STARS, the FAA and Raytheon have equipped a demonstration van that tours with a working version of full-service STARS.

Due to the success of the acquisition process and the team's outstanding effort, they were awarded the FAA's Office of Research and Acquisition's Sixth Annual Award for "Efficiency of the National Air Space."◆

The Ogden Air Logistics Center Develops Software That Automates the Minuteman III Messaging System

Pamela Bowers
CrossTalk

Without the Higher Authority Communications/Rapid Message Processing Element (HAC/RMPE) developed by Detachment 1, Ogden Air Logistics Center (ALC), the Minuteman III missile crews would be forced to resort to manual message decoding and processing. Instead, Detachment 1's software automatically collects communications data and displays it on computer screens, including duplicate suppression, error correction, and message formatting. The software reduces errors in incoming message formatting and speeds up processing.

The software program maintained by Detachment 1, Ogden Air Logistics Center (ALC) to support the Higher Authority Communications/Rapid Message Processing Element (HAC/RMPE) automatically codes and passes information for the Minuteman III missile crews into a Weapons System Control Element, which has the computer system that fires the missiles. Detachment 1 of the Ogden Air Logistics Center codes the software based on requirements from the Joint Chiefs of Staff and the Commander in Chief, U.S. Strategic Command. This allows the Minuteman III missile crews to receive changes that keep the missiles using the same Single Integrated Operational Plan (SIOP) as the manned bombers and Submarine Launched Ballistic Missiles (SLBMs).

Formerly, those change messages had to be manually handled. They were received over various communications systems printed out in the capsules, then processed individually by hand. The HAC/RMPE software collects incoming messages and displays them on the missile crews' computer screens, including conducting duplicate suppression, error correction, and message formatting. The crews are then able to do any alterations necessary and automatically feed the information into a Weapons System Control Element, the computer system that fires the missiles.

The HAC/RMPE software reduces errors in incoming message formatting and speeds up processing. No operational time has been lost due to failure in the system. "It is a project that successfully handles unpredictable volumes of changing requirements," notes Capers Jones, a Top 5 judge. "It received very high usability scores from users and had very good user satisfaction."

Another judge agrees. "This project had a perfect customer rating and appeared to perform flawlessly in all important customer respects," says Watts S. Humphrey.

These accomplishments were made despite the fact that the HAC/RMPE operational system is very antiquated, and severely limited in available memory for the additional functions. Yet, despite this limited memory, the Detachment 1 engineers and programmers, led by Toni Estes, Programming Team lead, have never had to decline a new requirement based on technical limitations. In fact, just last year they developed new techniques to allow even more HAC/RMPE messages to be utilized.

"It [HAC/RMPE] … successfully handles unpredictable volumes of changing requirements … and received very high usability scores … and had very good user satisfaction."

– Capers Jones
Top 5 Judge

The Delivered Product

Staff loyalty is the big thing that contributes to the project's success, stresses Capt. David Selnick, detachment commander. "I can't emphasize that enough. It's a high-pressure environment with short deadlines." In fact, he says that some of the equipment is so old, they are the only ones in the country still using it; mechanical upkeep is time consuming and difficult as no commercial/private sector languages are used in the operations.

Selnick adds that the team does not get predefined requirements documents. They have to figure that out themselves. "We get information on the fly like everyone else in the SIOP community. We determine the effect on the missiles, and what the software has to do to meet it – all within weeks of the deadline."

Despite the many professional drawbacks to working on this project, Selnick credits the "esprit de corps, importance of the mission, and personal dedication to that mission" for job retention. "More than half of our employees have been here since before 1997 – proving that people are not just marking time or counting the days until they can move to a more marketable position."

Estimation efforts are made based on research, design, and coding time alone. Size is not a factor unless the change request being considered would require alteration of an extreme number of files or use an excessive amount of system memory when operational (since the HAC/RMPE system has very little memory on which to draw). In that case, an estimate of memory usage would be made based upon the amount and type of data to be stored.

A Unit Test Procedure Sheet (UTPS) is used to document all steps that will be taken to test the change. It also doubles as a record of the actual test, as each test step is presented in checklist format.

"David Shaw and SrA Joshua Babcock comprise the detachment's software testing team. They use a system Test Procedure Sheet (TPS), which is similar to the UTPS on on a system-wide level. They also update the electronic TPS database, which was created to reutilize similar test procedures as well as provide a history in case the entire system ever needed to be re-qualified. The test report includes the completed TPS form, as well as written documentation of everything that occurred during system testing, including any new or pre-existing but undiscovered problems. Diane Moen, configuration manager, then releases the Software Version Description to highlight differences between the last release and the current one."

Reliability and Quality

While the technical challenge of this project appears to be typical, the reliability and quality parameters dictate otherwise. "The operational issues and highly sensitive nature of the application appear to make this a demanding technical project," says Humphrey.

Testing is performed on a HAC/RMPE console that is identical to the consoles in the missile capsules, as well as a simulator for a related system called the Weapon System Control Element (WSCE), which is also located in the capsules. Other test equipment includes a message generator that can mimic message traffic from any of the three communications platforms with which HAC/RMPE is designed to communicate, as well as several PCs and two protocol analyzers. Humphrey also gave the ALC high marks in quality assurance. "The Ogden process is comprehensive and the activities described are important. The broad use of measurements is impressive and the organization appears to be following a well defined and stable process."

First, peer reviews are conducted for every product produced. These are documented, and metrics are kept on number of defects, type, and rework time. Second, the configuration management program ensures that all release products are monitored, tracked, and documented throughout the entire software development process.

TSgt Scott Sorenson, the software quality assurance (SQA) representative, regularly audits the products for compliance; recommends changes or improvements; and keeps work time, requirements stability, and other relevant metrics. The software process improvement team, which meets as often as needed but at least quarterly, addresses issues that will enhance the simplicity and effectiveness of the software process. This team has at least one representative from every employee work area (process management, programmer/engineering, CM, SQA, and testing) to ensure that everyone's point of view is considered.

Capt Selnick, a Project Management Professional certified by the Project Management Institute, provides project oversight. Finally, a combined design review is performed with representatives from General Dynamics who are working on a version of the HAC/RMPE system to be used with a new type of survivable radio communications system. This combined review ensures that nothing "slips between the cracks."

Photo left to right, front row: *Paul Tomaso (General Dynamics), Toni Estes, Cindy Black, Diane Moen, Barbara Ronsick, Capt. David Selnick*. Back row: *TSgt. Scott Sorensen, SrA. Ian Swang, David Shaw, Edward Reed, SrA. Joshua Babcock, Ron Heacock (General Dynamics).* Missing from photo: *TSgt. Curtis Anderson and A1C Aaron Theisen.*

When determining its effort metrics throughout the process, the detachment defines its versions of cost performance index (CPI) and schedule performance index (SPI) in a manner that best suits their needs. When measuring CPI, cost is assessed in terms of man-hours only. This is similar to the traditional definition of SPI. Goal is 1.0. CPI = 0.79

The interpretation of this is that the estimate was within acceptable tolerances – due to the high volatility of the team's work, anything between 0.75 and 1.2 is considered within control. Capt. Selnick explains that the introduction of late requirements and the deletion of existing requirements at the last minute frequently play havoc with this metric. (Detachment 1 must account for actual hours expended on tasks that were not originally planned for, and it must discount hours spent on tasks that the customer decides at the last minute they do not want).

Regarding SPI, the percentage difference between planned and actual completion dates are computed slightly different than the traditional definition of SPI. Detachment 1 assesses the percentage difference between planned and actual in terms of calendar days. It calculates the length of time from project start to the actual milestone date, and divides it by the length of time from project start to estimated milestone date. In order to get a percentage difference, subtract this number from 1, and multiply by 100. This is a much more important measure to Detachment 1 than is CPI, because its end date is non-negotiable. Therefore, it can tolerate more variation in man-hours than it can in actual date slippage. This metric is calculated at three major milestones: delivery of SIOP Software Specification Matrices, delivery of engineering version of software to The Boeing Company, and delivery of final product to the field. The goal is zero or higher. Positive variation (delivering early) is good; negative variation (delivering late) is bad. All of last year's numbers were either zero or positive (on time or early).

The Cost per Stage is measured in man-hours. The different stages of the process are assessed in terms of their overall contribution to the total cost of the release. This metric does not include higher-level testing, since these dates and workloads are fixed by external agencies, and the team has little control over them. This historic data allows it to improve its estimation process.

What is the result of all these efforts? Operators have never encountered an error that would require them to stop using the software. No mission time has ever been lost due to a HAC/RMPE software error. SIOP interdependence means the software release date cannot be missed. While mandatory requirements are often introduced or changed months after they are supposed to be finalized, Detachment 1 has never failed to meet a date, and, in fact, often releases early. Required software functionality has never been reduced in order to meet the deadline.◆

The F/A-18 Advanced Weapons Lab Successfully Delivers a $120-Million Software Block Upgrade

Pamela Bowers
CrossTalk

As the F/A-18 Hornet becomes the Navy's nearly exclusive strike fighter, the Advanced Weapons Laboratory (AWL) steps up to the task of delivering a major software block upgrade. The software, called the 15C System Configuration Set (SCS), provides advancements that upgrade the interface between the aircraft mission systems and the aircrew. The AWL successfully delivered "real time" processing in an extremely mission critical system that pushes the technology envelope, and that requires absolute safety of flight.

The F/A-18 Hornet is the Navy's premier strike fighter, which now forms the core of the Navy's air warfare capability. As older aircraft are phased out of inventory, and the newest variant the F/A-18E/F is phased into the fleet, combat aircraft on the Navy's carrier decks will consist almost exclusively of F/A-18s. It is truly the heart of naval carrier aviation. The F/A-18 also serves as the primary fighter with seven U.S. military allies.

Success in today's air combat arena is a function of many variables. One of the most important is aircraft mission systems and their interface with the aircrew, especially in an era of exponential improvements in digital technology. The F/A-18 Advanced Weapons Laboratory (AWL) delivers these improved warfighting capabilities to the fleet.

As a full life-cycle activity, the F/A-18 AWL provides mission-system-engineering support for F/A-18E/F, as well as life-cycle support for out-of-production F/A-18A/B/C/D aircraft. The AWL coordinates F/A-18 system upgrades and enhancements and provides systems engineering for F/A-18 hardware and software. It accomplishes every aspect of the life cycle of the system configuration sets (SCS), including the software design for the mission computers and the stores management system. For the E/F aircraft, the AWL acts as system engineers and performs test activity; their teammate The Boeing Corporation is the design agent. Additionally the AWL manages a wide range of avionics and weapon systems developments, weapons integration, and foreign military products.

The F/A-18 AWL develops its own simulation laboratories, test equipment, and flight instrumentation; it generates and manages aircraft modification proposals and flight clearances. In its six integration and simulation laboratories, the AWL performs detailed subsystem and integration tests. The F/A-18 AWL and their Boeing teammates are Software Engineering Institute Capability Maturity Model® (CMM®) Level 4 software facilities. The AWL is well on its way to Level 5.

> **"This is a very large, real-time operational system that has made significant improvement in cost, schedule, and quality."**
>
> – Dr. Jack Ferguson
> Top 5 Judge

"The developers' transition to CMM Level 4 has resulted in reduced rework and reduced costs of test points," says Gary Kessler, Naval Air System Command representative. "The fleet is ecstatic."

Functioning as part of a greater F/A-18 Integrated Product Team (IPT), the people of the F/A-18 AWL are a Navy/industry team whose major contractors are The Boeing Corporation, Raytheon, and many other prime and support contractors. From technical leadership to business and financial management, they provide progressive, experienced management expertise for all levels of programs across a wide variety of disciplines.

Scope of the Project

During the top five contest award period of January 2000 to June 2001, the AWL delivered to the operational testers (VX-9) a major software block upgrade called the 15C SCS. This was approximately a $120-million effort that incorporated more than one hundred requirements. Here are just a few of the major products implemented in the SCS: the Joint StandOff Weapon, the AIM-9X Sidewinder, the Joint Helmet Mounted Cueing System, the Multifunctional Information Distribution System, the Digital Communication System, and the requirements from six foreign military sales customers.

"The 15C SCS effort was long and complex," says Boeing Block Captain Doug Garrette. The project began in the first quarter of 1997. The initial plan consisted of three builds with 61 USN statements of requirements (SORs) and 14 Foreign Military Sales SORs, he says. It grew to four builds and picked up 59 impact statements (additional requirements).

"The SCS involved the integration of three new weapons, five new avionics systems and a new aircraft configuration (A+)," says Garrette. Each of these programs was driven by their own schedules and needs, he adds. "15C had to be flexible and react to the dependencies that were brought on by these parallel activities. It was through the dedicated effort of the combined USN/Boeing team that commitments were met."

Watts S. Humphrey, a Top 5 judge noted the vast scope of the project. "While the technology appears to be relatively standard, at least for the set of best projects, the size, complexity, and number of systems involved does represent a significant technical challenge in itself."

In addition, the team was not co-locat-

[SM] *Capability Maturity Model-Integrated and CMMI are service marks of Carnegie Mellon University.*

ed but came from different organizations, says Barry Douglas, Advanced Weapons Laboratory, IPT lead. "But that didn't matter," he says. "The team pulled together from the beginning, overcame development difficulties posed by their separation, and produced a successful product.

The aircraft has more than 10 million words of code in more than 40 different processors. Each aircraft type has two distinct configurations. The major differences include the stores management computer (Q-9 or AYK-22), multiplex bus architectures (either five or six), radars (APG-65 or APG-73), two variants of the AYK-14 mission computer, and various other minor differences. The airframes different processors are programmed in eight variants of assembly language, and in Ada, C, PL/M-86, and Jovial. The software development environment also uses Fortran, Ada, and C.

The majority of the effort was in the two mission computers, stores management set, and radar. The software development environment has more than 4 million source lines of code (SLOC) in unique software. The documentation contained the complete set of logistics elements that include the following: aircrew publications, maintenance publications, training, trainer updates, technical directives, and mission planning module software.

Methods to Ensure Quality

The mission computer software team's effort was larger and more complex than most members had ever experienced, notes Kim Brestal, Boeing software lead. "The task included implementation of an extraordinary number of requirements representing new weapons, new aircraft systems and a new aircraft configuration.

"The biggest challenge, by far, was providing for efficient use of critical mission computer resources to allow for successful implementation of all the requirements," says Brestal. "An MC resource team was formed to devise and implement risk mitigation plans for each affected resource."

Truly this project was large and complex agrees Capers Jones, a Top 5 judge. "The combination of low rates of delivered defects and high levels of customer satisfaction indicates this project was very well planned and managed." Jones cites the AWL's processes as a key to their success. "The project was produced by a SEI CMM Level 4 organization, and demonstrates the value of the higher CMM levels."

To achieve this quality goal, the AWL team performed the following:

- Achieved a CMM Level 4 and aggressively started moving to Level 5.
- Used the Capability Maturity Model®-Integrated^{SM} to assess organizational maturity and process area capability. Established priorities for improvement and methods to implement these improvements.
- Published, updated, and distributed a strategic plan that defines basic core beliefs, visions, and mission.
- Tested jointly with the Operational T&E Squadron throughout the verification phase of 15C. This gave them an early look at the product and gave the AWL earlier insight into operational problems in the product.
- Published an F/A-18 AWL Management and Systems Engineering Process Manual to systematically identify and apply leverage to areas of weakness and expand on what they do right.
- Maintained and improved its system-configuration review board process to obtain a very solid, well thought out, and adequately funded set of requirements.
- Improved on and used a comprehensive set of metrics. An example of the numerous metrics used is the indicator used to indicate software maturity level. At 0.12 software anomaly reports per test hour, the software is ready for operational test.

Results Show Success

The group not only produced the 15C SCS, but also was developing additional major SCSs, each at different stages, all at the same time, says Douglas. During the past 10 years, the AWL delivered four major F/A-18C/D SCSs as the total aircraft software increased to more than 10 million words. Each showed constant and unprecedented improvement. Considering 15C as the latest SCS, the following data apply:

- Reduced cycle time from 56 months to 38 months.
- Reduced schedule slips from 12 months to on time.
- Decreased rework rate from 20:1 to 3:1.
- Decreased regression testing from 70 percent to 20 percent.
- Decreased redundant testing from 100 percent to 10 percent.
- Improved test efficiency from 0.42 to 1.6 test points closed per hour of test time.

SCS 15C had the following specific indicators:

- Defect density was very low, 3.8 defects per KSLOC – down from 13.5.
- Productivity in the design phase was 3.45 man-hours per SLOC – down from 15.7.
- Design phase cost was $200 per SLOC – down from $725.
- Life-cycle cost was $400 per SLOC – down from $1,170.
- The number of test flights was 0.6 flights per KSLOC – down from 3.1.

Top Photo: *A VX-9 F/A-18 Aircraft Over the China Lake, Calif., range.* Bottom Photo: *The F/A-18 Advanced Weapons Lab located at China Lake, Calif.*

To date, the fleet has not reported any problems with SCS 15C. Likewise, the AWL has yet to receive any software trouble reports from the fleet on its similar product, System Configuration Set 13C, delivered three years ago.

"This is a very large, real-time operational system that has made significant improvement in cost, schedule, and quality," says Jack Ferguson, a Top 5 judge.

Accomplishments Are Applauded

For software of this size and complexity, the AWL feels this is one of the top software projects in the government for total life-cycle costs, quality, schedule, and performance. It says this is especially commendable considering this is "real time" processing in an extremely mission critical system that pushes the technology envelope, and that requires absolute safety of flight.

If the high cost of flight test vs. the commercial process of free "beta testing" is factored out, this software is a bargain in any commercial market, says Douglas. "The overall cost and quality statistics for this level of effort are truly outstanding, but the improvement during the past 10 years is truly phenomenal."◆

The Bureau of the Census Delivers the First System to Use Digital Imaging Technologies to Process Forms

Pamela Bowers
CrossTalk

For the first time ever, the Bureau of the Census (BOC) used imaging and recognition technologies to process forms resulting in more data being received faster than ever before. The BOC and Lockheed Martin successfully developed a system that automatically processed more than 150 million multi-page Census forms in 170 days. In the end their data accuracy was exceptional, reaching 99 percent.

The Bureau of the Census (BOC) contracted with Lockheed Martin Mission Systems to deliver an imaging and recognition system that would automatically process more than 150 million multi-page Census forms in 170 days. The delivered product was the DCS2000 system; a fully integrated system capable of logging and electronically reading census forms, storing the data on high fidelity backup tapes, and tracking the data via large, Oracle-based databases.

The DCS2000 program was an extremely high profile event with milestones and deliveries set by Congress. To miss these would subject the program, and the BOC, to a high level of scrutiny from Congress and the General Accounting Office, as well as the press. DCS2000 met all major milestones, delivering a high quality system exceeding all accuracy requirements. Notably, it was the largest, most accurate imaging and recognition program in history. The DCS2000 was the first Census using digital imaging, and the first handled by contractors.

The resulting system was deployed to four Data Capture Centers across the United States and began processing Census forms on March 6, 2000. Each Data Capture Center is staffed by approximately 2,000 people who collectively processed the equivalent of 1.5 billion pages of information in just 170 days – the largest data capture ever. The integrated system is capable of the following:

- Quickly checking in large numbers of U.S. Census forms.
- Electronically reading the data on the forms (known as Title 13 data).
- Storing the data in large, flat files that were shipped nightly to the BOC customer.
- Tracking the data and the forms movement through the system via large, Oracle-based databases.
- Providing a keying function for error correction.
- Backing up data to tape.
- Ensuring that forms could be shredded at the conclusion of the processing (with confidence that no data was lost).

The system was developed using a "cluster concept" that allowed for proper system scaling (depending on BOC needs). A cluster included three high-speed scanners and all peripherals needed to support those scanners. In total, 33 vendors were brought into the solution and integrated into the DCS2000 system.

> **"A solid methodology of communications was adopted to ensure that problems were addressed at the appropriate levels."**
>
> – Brenda Zettervall
> Top 5 Judge

"This project was technically challenging and completed quickly," says Capers Jones, a Top 5 judge. "It made use of new technologies and also was required to process an extraordinary volume of data when deployed."

Development and Use Environment

The DCS2000 System was developed at the Bowie Computer Center in Bowie, Md. (a customer location that houses the BOC computer facilities). The program used a spiral development model and developed the custom code in C++ on a Windows NT platform. The two major databases (status and management) were developed using Oracle.

The architecture used on the DCS2000 program allowed for a multitude of changes to occur without changing the fundamental design. It was expandable, so as requirements increased, the system was able to get larger without a redesign.

Adherence to a well-defined software development process was a must for an effort of this magnitude. Lockheed Martin Mission Systems was recently certified Software Engineering Institute's Capability Maturity Model® Level 5 for software development, in part based on the independent assessment of DCS2000 processes and procedures.

A multi-functional lab installed in the Bowie Computer center with the following environments: development, software integration and test (SWIT), and system test. A configuration management department ensured that the baselines for each environment was up to date.

"A solid methodology of communications was adopted to ensure that problems were addressed at the appropriate levels," says Brenda Zettervall, a Top 5 judge. She noted that the BOC technical staff became members of the Integrated Product Teams during the development and test period. As such, the BOC had detailed insight into the direction that each technical product was taking and was able to influence key technical decisions. At the management level, daily meetings were held between the DCS2000 Program Manager and the BOC counterpart. Regularly scheduled executive meetings were also conducted at the director and vice president level with their BOC counterparts. In addition, Zettervall says that the program had a robust metrics process that identified problems early (before they became large and unwieldy).

"The biggest contributor to the project's success was the complete openness between the BOC customer and ourselves," says Bill MacDonald, program manager, DCS2000. "There was nothing kept back. If we had a problem, the customer was part of the integrated development team that met daily."

In fact, three separate stand-up meetings were held daily, says MacDonald. There was a morning program management meeting, next came a midday teleconference, followed by an evening roundup meeting, explains MacDonald. "We functioned as a cohesive team," he says. "They had complete confidence we were telling them everything."

The final system was installed at four data capture centers located in Baltimore; Jeffersonville, Ind.; Phoenix, Ariz.; and Pomona, Calif. Each data capture center had a high-speed link connecting it to the central technical support function at the Bowie Computer Center.

Changes Made, Quality Maintained

The DCS2000 program went through a number of requirements changes prior to delivery. Twenty-nine contract modifications totaling $170 million were negotiated during the development and support of the program. On many occasions the change requests were the result of congressional action. For instance, the BOC had originally planned to use a statistical sampling technique for distribution of forms. That decision was reversed and therefore a change request was submitted to accommodate a traditional census.

Some change requests were made to mitigate risks jointly identified by Lockheed Martin and the BOC. For instance, a risk to the timely completion of data capture was identified whereby, if the production keyers did not reach a certain keying rate, data would not be available to report to the president of the United States by Dec. 31, 2000. To mitigate that risk, a major engineering change proposal was drafted, approved, and implemented four months prior to the beginning of data capture. This change allowed the system to process the data necessary for delivery to the president (called 100 percent data) and allowed a second pass to capture all other data.

There are two reasons for keyers being part of the system. First, not all marks and characters could be read with 100 percent accuracy. When a field did not fall within a predefined confidence level, it was sent to a keyer for validation. For instance, if the word was "Smith" and the system was unsure whether it was "Snith" or "Smith," a keyer would look at the field (electronically) and make the necessary entry. This was known as Key From Image. In addition, for those forms that were mangled, the system had a Key From Paper capability. This allowed direct entry into the system manually.

Photo: The *Lockheed Martin "Nova" team award recipients for outstanding contributions on the U.S. Census Program.*

The second reason for keyers was for quality control. Constant sampling was done whereby the data from a completed form was pulled, the form sent to a keyer, and then routed to another keyer. The data was then compared to the original electronic processing. If all three matched, then the electronic processing was successful. This was all done without keyers knowledge, i.e., the keyer did not know whether they were processing a field that was low confidence, or processing a field for quality control. This allowed the BOC to assess the system accuracy.

All engineering change proposals (ECPs) were completed on or ahead of schedule. Metrics for the DCS2000 program were collected and presented to program management and the customer on a monthly basis. A Schedule Performance Index (SPI) of 1.07 and a Cost Performance Index (CPI) of 1.0 showed a program that was ahead of schedule and on budget. Other metrics collected and presented showed a steadily improved defect rate for development, Software Integration and Test (SWIT), and systems integration (SI) test.

A robust, automated quality assurance (QA) process that measured the quality of the imaging and recognition system was built into the DCS2000 system. This process allowed for continuous monitoring of system performance/accuracy and allowed the DCS2000 team to do additional tuning for greater accuracy. In fact, the accuracy of the data sent to the BOC exceeded all expectations. Specifically, results independently measured by Rochester Institute of Technology indicated 99.89 percent data accuracy for optical mark recognition and 99.4 percent data accuracy for optical character recognition.

This program was a highly visible development effort with a congressionally mandated schedule that could not slip. An open style of communication was established with each vendor, which led to a true teaming concept. Each company understood their part and considered the success of the Census paramount to decision making (as opposed to what was best for an individual company). This attitude strengthened the final product and made it a truly integrated system.

"This project team arranged a great partnership and reacted well to customer direction," says Alan Berlinger, BOC Data Capture program manager. "They started with a firm deadline, but ambitious requirements. They ended with … the largest data capture effort ever."

"Never before has the BOC processed so much data so quickly," says MacDonald. "When you consider the magnitude of this program, the congressionally mandated milestones, and the large number of changes to the baseline, it is easy to see why nothing short of a true partnership between government and private sector could make this work."◆

The TRW Tactical Systems Division Builds the Next Generation of Tactical Army Operations Systems

Pamela Bowers
CrossTalk

Revolutionary changes in warfare are possible with the Force XXI Battle Command – Brigade and Below (FBCB2) project built by TRW. It is a system of networked computers, radios, and communications systems that provides soldiers in the field with situational information that allows them to be as effective as possible in conducting their mission. Users operating the system know what to do, where to go, what surrounds them, how to avoid danger, and more.

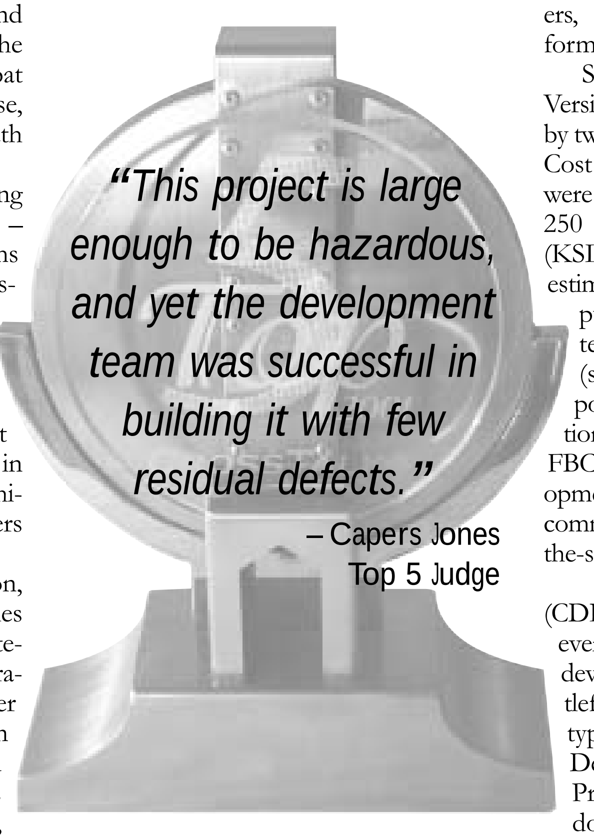
The Force XXI Battle Command – Brigade and Below (FBCB2) project provides breakthroughs in the effectiveness of the tactical Army. The system is an over-the-air network of computers, radios, and communications systems that enables the Army to utilize knowledge for combat advantage. FBCB2 is designed for intense, dangerous, conditions with life-or-death consequences for Army forces.

The primary users are soldiers ranking from private to colonel, possibly higher – primarily operating their FBCB2 systems within vehicles. They interact with the system using a touch screen and a graphical user interface designed to be tolerant of strong vibration and temperature. The users also include a class of specialized personnel doing network management and other executive functions situated in tent complexes or special command vehicles known as Tactical Operations Centers (TOCs).

In a typical brigade (or larger) mission, FBCB2 is hosted on hundreds of vehicles (exercises have been run with approximately 1,000 vehicles). Users are in collaborative, near-real-time contact with each other as the system shares location information called situational awareness (SA) data and command and control (C2) messages (e.g., orders, descriptive map overlays, logistics requests, alerts/warnings, and status reports). Users employ the system to know what surrounds them, what to do, where to go, how to avoid danger, etc. Frequently the system is operated "on the move," including at night or in poor visibility conditions. FBCB2 also includes planning tools to help a commander plan and analyze a mission.

FBCB2 is proclaimed by the Army customer and users as a "home run" in next-generation operations and a revolutionary change in warfare in these ways:

1. Expanding the range of operations.
2. Reducing reliance on already scarce voice communications availability by providing position information digitally and automatically.
3. Coordinating maneuver at night, in bad weather, or during times of reduced battlefield visibility.



> "This project is large enough to be hazardous, and yet the development team was successful in building it with few residual defects."
>
> – Capers Jones
> Top 5 Judge

4. Minimizing the commander's unknowns in his decision cycle.
5. Reducing fratricide.
6. Increasing the force's lethality.

Complete Source Code Control

The challenge to FBCB2 software development was to build something over which complete control of the source code was maintained. Since the Army is buying about 60,000 platforms, it did not want to have any significant licensing costs to drive up overall deployment costs.

TRW is the prime FBCB2 contractor and provides project management, engineering, software development, systems integration, and test and life-cycle support. Raytheon is a major teammate on FBCB2 for almost half the software, especially SA and C2. TRW also oversees the development, test, and fielding of FBCB2 computers, communications hardware, and platform installation.

Software size estimates history for the Version-1 FBCB2 proposal was generated by two parametric cost models: Constructive Cost Model (COCOMO) and Revic. These were calibrated to TRW projects estimating 250 thousand source lines of code (KSLOC). Subsequent FBCB2 versions' estimates for costing, planning, and tracking purposes were based on a more modern technique of "required capabilities" (somewhat analogous to function points), not SLOC. The relative proportions of the present Version (v3.4) of FBCB2 are approximately two-thirds developmental, and the balance a mixture of commercial off-the-shelf, government off-the-shelf and non-developmental items.

The Contract Data Requirements List (CDRL) requirements are comprehensive, even though FBCB2 is an evolutionarily developed and incrementally delivered battlefield system. Twenty eight document types ranging from the Software Development Plan (SDP) and Software Product Specification (SPS), requirements documents, Interface Control Documents (ICDs), and architecture (and other) trade studies to software test, quality plans, and user training materials have constituted more than 900 version deliverables and approximately 110,000 pages. All deliveries are now in electronic form.

The FBCB2 software development environment matches the customer's principal target platform – Intel computers running Posix-compliant Unix (Solaris) – for the advantages of more realistic early testing and reduced risk by avoiding cross-compiling and building. As cost savings, developers have Windows PCs on their desktops (running Microsoft Office for office productivity functions), with network log in to the Unix machines, which comprise the compilation and test environment.

Compilers are freeware Gnu compilers. The X-Designer tool is used for screen designs and rapid, reliable template code generation. RTM is used for requirements management and test traceability. FBCB2 employs an integrated configuration management and software problem report process, and has recently migrated its tools support to Rational ClearCase/ClearQuest from earlier tools because of more efficient support for multiple baselines, with the added benefit of metrics collection and analysis.

Three operational usage environments (operating system/hardware configurations) cover the 40-plus types of Army vehicles and aviation platforms upon which FBCB2 is installed: Intel machines running Solaris, Sun SPARC machines running Solaris, and Intel machines running LynxOS. The first Intel/Solaris environment was selected for cost effectiveness, given more than 9,000 initial FBCB2 installations. The other two are the environments in existing systems into which core FBCB2 functionality is being embedded.

Customer Satisfaction Equals Quality

More than 60 FBCB2 software deliveries have been provided to exercises, training commitments, and demonstrations providing the ultimate proof of quality: high user satisfaction with no major problem reports. Instead, FBCB2 receives increasing demand for capability expansion beyond contract scope.

User evaluation and feedback at every stage of product evolution is a primary FBCB2 quality method. FBCB2 also employs quality processes and procedures that ensure delivery of verified products within predictable cost and schedule. These processes are based on TRW's quality systems that exceed ISO 9001:2000 and the Software Engineering Institute's Capability Maturity Model®.

Management proactively monitors the project and conducts risk management using metrics-driven decision making. Monthly cost/schedule-variance reviews focus on individual task achievements vs. planned schedule and cost budgets to assure contract programmatics' satisfaction. Monthly metrics reviews emphasize quality factors such as defect density, key system performance parameters, schedule satisfaction of derived task-level activities, tracking of critical-activity paths, staff leveling, and measuring compliance with project/contract plans.

FBCB2 has introduced advanced statistical process control methods to better identify problem areas and exemplary subprocesses for process improvement. A near-daily software Configuration Control Board (CCB) meeting expedites field/test recommended changes; a higher-level project CCB controls baselines in project configuration management and enforces the disciplined boundary between development and formal integration and testing.

Image: *FBCB2 can be used in a vehicles to gain information superiority and a tactical advantage.*

Capers Jones, a Top 5 judge commended the project's processes when he said, "This project is large enough to be hazardous, and yet the development team was successful in building it with few residual defects. The size of the project was stated to be about 2,300,000 SLOC or roughly 22,000 function points. Project management and both risk and quality control on this project were extremely proactive."

Good Field Ratings

No major deficiencies have been reported in scheduled product deliveries, i.e., there has been no leakage of significant defects from final, customer-witnessed formal testing to delivered products. Because of consistent, flawless delivery of required functionality, and the revolutionary character and break-through capabilities of the FBCB2 system, most of the reports received from the field (dozens) are "good ideas" for enhanced functionality beyond contract requirements, not problem reports.

The bottom-line implication is that all substantive problems in contracted software capabilities are detected and fixed by FBCB2's multi-level developmental test processes before fielding.

Many field events attest to FBCB2's revolution in ground warfare. Following is a comment from the project's Division Capstone Exercise: "On 2 April, tanks and other heavy armored equipment were able to ignore a blinding sandstorm and cause the opposing force (OPFOR) to lose 60 percent of its combatants," said Col. John Antal, exercise chief of staff. "You couldn't see your hand in front of your face. On the FBCB2, that's the computer system on the tanks and the Bradleys, the friendly forces knew where they were," he said. "They didn't fire artillery at themselves. They know what the [terrain] obstacles were and they had a good read on the enemy," ... thus defeating a usually invincible OPFOR.◆

Photo: *Part of the Main FBCB2 Team at TRW's Dominguez Hills Technology Center.*

CROSSTALK Honors the 2001 Top 5 Quality Software Projects Finalists

Pamela Bowers
CROSSTALK

There were so many successful government projects entered in the first Top 5 Quality Software Projects contest that it was difficult to narrow the field. As a result, the following 11 projects are being honored as 2001 Top 5 Finalists. A brief description of each project is included here. Look for a more detailed article on many of these projects in upcoming CROSSTALK *issues.*

CWS – Command and Control Mainframe System

Customer: Air Force Satellite Control Network, Command and Control Segment Sustainment

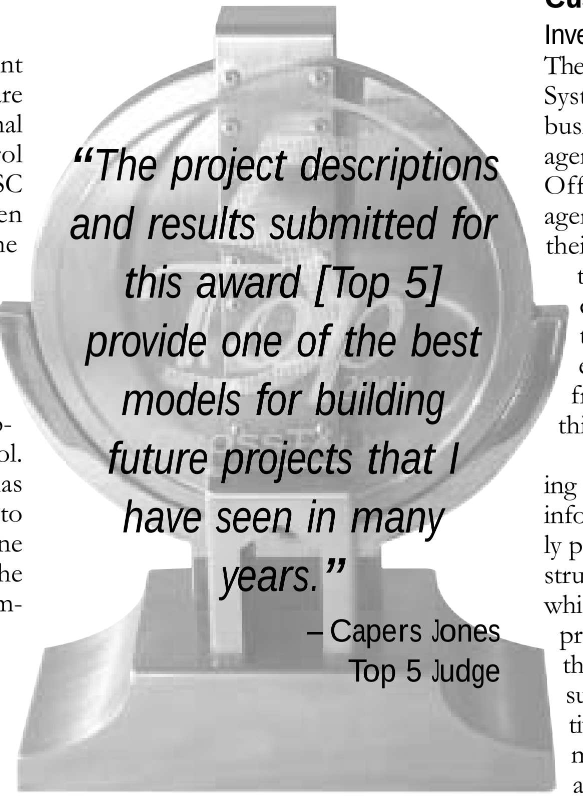
The Command and Control Sustainment Contract (CCSC) team delivers software fixes that allow the Air Force operational community to command and control satellites at a high success rate. The CCSC team delivered 25 products between January 2000 and June 2001 to ensure the success of 14 unique satellite control complexes of the Air Forces Satellite Control Network. These deliveries included non-scheduled emergency software deliveries as well as scheduled software maintenance deliveries in support of satellite command and control. For this time period, the CCSC team has not introduced any priority problems to the operational baseline for any baseline or emergency products delivered to the operational command and control complexes.

Industrial Automation Automated Testing Software Section/F100 Windows Intelligent Trending and Diagnostic System

Customer: Information Resources Branch of the Propulsion Directorate

The Windows Intelligent Trending and Diagnostic System (WITADS) software analyzes performance data of the F100 engine to predict impending problems, reduce diagnostic time on the flight line, incorporate corporate knowledge into the software, and utilize current field equipment. WITADS analyzes downloaded jet engine in-flight data to determine health, faults and/or alarm cautions associated with the engine, and provides a prediction of future engine faults or cautions.

The field service evaluation (FSE) process was utilized to determine the operational/objective status of the program. For example, FSE documents reported that ITADS was identifying possible failure modes before failure. This saved additional damage to the engine by catching the possible failure and preventing damage down stream. For example, the annual cost avoidance for damaged F100 engine afterburner exhaust nozzles is approximately $1.5 million.



> "The project descriptions and results submitted for this award [Top 5] provide one of the best models for building future projects that I have seen in many years."
>
> – Capers Jones
> Top 5 Judge

The F100 engine is used in the F-15 and F-16 fighter aircraft, thus two versions of the software are required. WITADS is an artificial intelligence system using a rules-based expert-system methodology derived from the technical knowledge of Air Force (AF) engineers and engine manufacturer specialists. WITADS is designed to interface with the standard ground station software used by flight-line technicians at F-15 and F-16 AF bases worldwide.

Investigative Information Management System Program Management Office

Customer: Air Force Office of Special Investigations

The Investigative Information Management System (I2MS) is the only activity-based business workflow and information management system that provides Air Force Office of Special Investigations field agents with the ability to capture all of their investigative data in one place and then generate the final product for the customer. It is a true workflow system that follows the investigator through every aspect of criminal investigations, from murder to espionage and everything in between.

I2MS is a user friendly means of tracking and saving all collected investigated information with reports being automatically populated and published. The I2MS data structure allows for link analysis on the fly, which means an investigator will have all previous reports and incidents related to the suspect at his or her fingertips during subsequent investigations. Also, investigative leads that previously required days of mail-time and hours of briefings with assisting investigators are now conducted completely within the confines of the database. The organization estimates it will recover its costs within three years of operation in addition to the increase in capability and the reduction of missed investigative steps due to human error.

Lockheed Martin Mission Systems All Source Analysis System

Customer: Project Management Office Intelligence Fusion

The All Source Analysis System (ASAS) program produces a family of intelligence analysis software products. These products provide intelligence analysts common applications, communications management and message processing, and situation awareness/development tools to create an accurate and timely common picture of the battlefield. The system has passed its developmental and operational testing and is, in fact, operational and deployed worldwide and is in the hands of soldiers in the field today. Army intelligence analysts in the Balkans are using the system to provide critical intelligence in support of their ongoing mission. Its true value may only be measurable in terms of mission accomplishment or lives saved in future actions.

The systems have been engineered to support maximum interoperability and flexibility for the intelligence community by providing Battlefield Systems Inter-faces, accommodating other Battlefield Functional Area clients and utilizing government off-the-shelf and commercial off-the-shelf software. Twenty-two deliveries occurred in the last 18 months with all 97 contract data requirements lists delivered on or ahead of schedule. The ASAS Block II Remote Workstation software is the intelligence and electronic warfare component of the Army Battle Command System.

Minuteman Automatic Test System for Launch Facility Operational Ground Equipment (MATSO)-OO-ALC/TISMB

Customer: LBM, Intercontinental Ballistic Missile System's Program Office Ground Systems Division

The Minuteman Automatic Test System for Launch Facility Operational Ground Equipment (MATSO) team provides high quality software support for the Minuteman and Peacekeeper Missile defense program under the goals of the Software Capability Maturity Model® (SW-CMM®). The TISMB provides an invaluable service to ensure that an aging defense system is maintained to peak performance. TISMB uses statistical methods such as control limits that have been established for cost and schedule metrics. This metrics indicates whether TISMB is in control of their processes, and where problems are likely to occur. Data is used to initiate process changes as needed. Defect data provides TISMB with the necessary information to perform a causal analysis on the defects found at peer reviews. This causal analysis not only removes the defect, but also prevents the recurrence of the defect in the future.

Using the best information available, indications are that process improvements have reduced maintenance schedule and costs. Software releases for TISMB have been virtually error-free. TISMB is able to produce a higher quality product at reduced cost. MATSO was a focus project of its parent organization's SW-CMM® Level 5 assessment and for the past several years has consistently rated Level 5 on internal quality assurance reviews.

NASA Glenn Research Center/Numerical Propulsion System Simulation

Customer: NICE-1 Consortium, comprised of NASA Glenn Research Center, General Electric Aircraft Engines, Pratt & Whitney, The Boeing Company, Honeywell, Rolls-Royce, Williams International, Teledyne Ryan Aeronautical, Arnold Engineering Development Center, Wright Patterson Air Force Base

Numerical Propulsion System Simulation (NPSS) V1.0 is an object-oriented preliminary and conceptual design code used by aerospace engineers to predict and analyze the aero-thermodynamic behavior of commercial jet aircraft, military, and rocket engines. NPSS V1.0 also allows the dynamic substitution of its engine components (objects) to components (objects) of greater fidelity; a concept called Numerical Zooming. It is written in C++ and was developed following a production software engineering process. NPSS was developed to reduce cost and risk and increase capability and accuracy by numerically creating aerospace propulsion systems before hardware is ever built. NPSS V1.0 is the first deliverable from this vision.

National Missile Defense Battle Management, Command, Control, and Communications

Customer: National Missile Defense, Joint Program Office, BMC3 Program Office

The National Missile Defense (NMD) system is comprised of weapon, sensor, and Battle Management, Command, Control, and Communication (BMC3) elements that provide the capability to detect, engage, and negate threatening inter-continental ballistic missiles. The BMC3 element provides the integrating mechanism for controlling and directing all aspects of the NMD system operations. TRW is responsible for developing and employing the battle management command and control, test exerciser, in-flight interceptor communications system, and communications node equipment/network system manager (primarily COTS integration). Since 1995, TRW has delivered five BMC3 product increments and successfully achieved all technical milestones for the 30-month NMD basic contract. TRW cost was $1.9 million under budget with less than 1 percent schedule variance.

Naval Oceanographic Office, Geophysics Fleet Mission Program Library

Customer: Commander, Naval Meteorology and Oceanography Command

The Geophysics Fleet Mission Program Library (GFMPL) is a rapid-response, on-scene, environmental prediction software suite used to quickly assess the effects of the environment on fleet sensors, platforms, and weapons systems. Products include electromagnetic/electro-optic propagation conditions and oceanographic acoustic predictions. The software applications in the library are used to increase safety for the warfighter and to increase his combat effectiveness.

The number of its operational customers and the number of complaints on software problems determine the measure of value for GFMPL. For example, if a customer uses the tides application to plan a ship's entry into a port and the ship arrives safely, then that is an important measure of effectiveness. If a Navy Seal uses the solar lunar application to determine moonrise and lumination factors for mission planning, and the application forecast is accurate, then the value is priceless. In 2001, GFMPL was sent to more than 330 operational customers with no complaints of software defects. The GFMPL release is on a fixed schedule of six months per iteration. Analysis of the last iteration indicated that process efficiency resulted in a 23 percent increase in actual versus planned requirements.

Ogden Air Logistics Center, Radio Solar Telescope Network Re-Host Software

Customer: Space and Missile Systems Center, DMSP/SESS Division, Software Branch SMC Det 11/CIDS

The Radio Solar Telescope Network (RSTN) re-host project will replace the computerized portion of the Radio Interference Measuring Sets (RIMS). RIMS is part of the RSTN and consists of eight radio frequency meters that measure solar emissions. The RSTN re-host system is a functional replacement for the HP-1000 computer system currently in use at four of the U.S. Air Force Radio Solar Observatories.

The HP-1000 uses Fortran and assembly language software routines to analyze and display radiometer activity to the user and compose solar weather messages to transmit to a central facility. By eliminating the older hardware and software, this project will save an estimated $160,000 per year paying for itself in the first year or two. In addition, this project will eliminate a number of maintenance problems and associated costs. Installation started a full month earlier than expected.

Other benefits include reduced training due to the re-hosts' graphical user interfaces and other user interfaces are more modern and similar to those commonly used by computer operators.

Reserve Component Automation System

Customer: Army Reserve Component

The Reserve Component Automation System (RCAS) is a $2.3 billion automated information system that provides the Army National Guard (ARNG) and United States Army Reserve (USAR) with the capability to administer and manage day-to-day operations and mobilization planning. Added complexity comes from working with these two major customers. RCAS links more than 57,000 personal computers at 10,500 ARNG and USAR units through a wide area network at over 4,000 sites in 54 locations: states, territories, and the District of Columbia. Each USAR unit reports to the governor and has both a state and federal support mission. It is noted that all users do not have the same needs or chains of command.

Beyond providing scalable system architecture, RCAS is an open system with two separate subsystems: classified and unclassified.

In March 2001, the project completed fielding of the system's infrastructure 18 months ahead of the original deployment schedule. In March 2003, delivery of all software functionality will be complete. Customer requirements span 11 functional areas and consist of over 70,000 function points of government off-the-shelf/commercial off-the-shelf and RCAS developed software.

SOF EISE IPT IDAS/MATT Upgrade

Customer: Special Operations Force Systems Program Office

The Interactive Defense Avionics System/Multi-Mission Advanced Tactical Terminal (IDAS/MATT) is a modification to the MH-53J Pave Low III (PL-III) aircraft and is now designated as PL-IV. The IDAS/MATT upgrade program incorporated the PL-IV aircraft system onto the PL-III simulation network. This upgrade makes possible the software maintenance of the operational flight programs of the MH-53M weapon system. The MH-53M with IDAS/MATT is the world's most software intensive and technologically sophisticated helicopter. The continued high Mission Capability Rate (five percent over Major Command goal) of this Force Activity Designator 1 weapon system is only possible due to the support rendered by the Special Operations Forces Extendible Integration Support Environment with the IDAS/MATT upgrade.

The simulation network now supports both aircraft configurations with minimum hardware reconfiguration required. Hardware changes included updating the user interface function to reflect PL-IV cockpit changes and addition of an Embedded Computer Systems/Line Replacable Unit (LRU) rack to host PL-IV unique LRUs. Software changes included the modification of 10 existing LRU simulations. In addition the flight, visual scene driver, and terrain/target simulations were modified.

Software block cycle change cycle time has dramatically dropped with the EISE upgrade. During a recent deployment, an emergency change request was analyzed and a fix developed, coded and tested in about two weeks.◆

Top 5 Software Projects Scoring Criteria

Reviewers from the Software Technology Support Center (STSC), Hill Air Force Base, Ogden, Utah, used the following criteria and point system to score all nominations in order to select the 2001 Top 5 U. S. Government Quality Software Projects. Each nomination was awarded points (up to a maximum value) based on how well the project performed within each category: customer value, performance, technical value, and reviewer's discretion. Each nomination was scored by at least three STSC consultants or engineers with the top one-third of nominations being scrutinized more closely by the internal board in order to select finalists.

Customer Value – Maximum 40 Points

Problem Reports

- Were responses to the problem reports and questions timely?

Value

- What was the measured value to the customer's mission (return on investment)?

Benefits and Satisfaction

- Is the end product useable?
- Is the customer satisfied with the end result?
- What other benefits were provided to the customer?
- Was the developer collaborative?
- Did the developer listen to the customer?
- Was the developer knowledgeable? Informative? Helpful?
- Was the developer professional in letting the customer know requirements tradeoffs?

Performance – Maximum 25 Points

- Did the developer meet the contracted schedule?
- Did the developer meet the contracted budget?
- How many problem reports have been written against the product since system test?
- Is the customer satisfied with the end result?

Technical Value – Maximum 20 Points

- Was the problem challenging? How hard was this project to implement?
- Was the solution innovative? What approach was used to solve the problem? What technical value did they provide to the world?
- Is the project reusable? Can someone else use the end product, portions of the end product, code, process, or the product's technology to solve a future government problem?
- Is the project repeatable? Given a similar problem, could this group repeat this success or were they just lucky this time? (Did they use defined processes, trained people, etc.)

Reviewer's Discretion – Maximum 15 Points

Use or don't use these points as discretion dictates. Suggested considerations include the following:

- Previous awards (CMM, ISO 9000, Malcolm Baldridge, etc.).
- Customer. (Will one small organization use this or will it be dispersed worldwide?)
- Do they have measures that can be used for oversight and additional improvements?
- What is the atmosphere/morale of the developing organization?

Government Software Projects Rank High in Major Critical Success Factors

Capers Jones
Software Productivity Research Inc., Artemis Management Systems

Capers Jones, one of the judges for CROSSTALK's first "Top 5 Quality Software Projects" awards, compares the projects he reviewed for the contest with those he has recently seen embroiled in legal disputes in the "real world." He comes away refreshed to see proof in these government projects that large and complex software projects can be finished on time, within budget, meet with favorable user reactions, and have few remaining defects after delivery.

Having just returned from working as an expert witness in a software breach of contract lawsuit, it was very refreshing to read all 16 of the finalists' project descriptions submitted for the evaluation for CROSSTALK's Top 5 Quality Software Projects awards. None of the 16 projects made the kinds of mistakes in project management and quality control that have kept me in various courtrooms during the past few years.

Since most of the software projects that end up in court exceed their cost estimates by several hundred percent, have distressingly poor quality control, and bungle project management tasks such as sizing and schedule planning, it was quite enjoyable to read how competent software vendors go about building successful packages. All 16 teams are to be congratulated. In particular, teams with projects larger than 10,000 function points or 1 million single lines of code (SLOC) deserve a great deal of credit.

Software projects are influenced by more than 100 different factors. However, when similar projects are examined where one is successful (i.e., on time with good quality) and one is a failure (i.e., cancelled, delayed, or inoperable) about a dozen key factors tend to distinguish success from failure. Since there are many more large-project failures than successes, these were a rare breed.

The accompanying sidebar lists major factors associated with both success and failure that I have noted in examining many thousands of software projects. The list is taken from my book *Patterns of Software System Failure and Success*, International Thomson Computer Press 1995.

Of the 16 finalist projects submitted, all were better than average in every one of these critical factors. The top five projects ranged from "very good" to "outstanding" in all of these critical factors.

Another observation from reviewing the results of the nominations is the solid evidence that ascending the Software Engineering Institute's Capability Maturity Model® up to Level 3 or higher is well worthwhile. Indeed, for really large applications in the range of 10,000 function points or 1million SLOC, Level 5 is the desirable level for optimal performance.

There are so many software overruns and outright cancellations that it was quite refreshing to see "existence proofs" that large and complex software projects can be finished on time, within budget, meet with favorable user reactions, and have few remaining defects after delivery.

This evaluation of top software projects is so useful that I think CROSSTALK should be commended. My personal hope is that similar evaluations of top software projects will continue to be carried out annually.

It is hard to learn much from average projects. We can learn things from failures and disasters, of course, but we do not want to pattern our own projects on unfortunate models. The project descriptions and results submitted for this award provide one of the best models for building future projects that I have seen in many years. There are hundreds of ways to botch up projects, and only a few ways to build them successfully. Excellence in project management, estimating, measurement, quality control, and change control are all required for successful results. All of the projects submitted are to be commended, and the top five deserve accolades from the software community.◆

Successful Projects

- Effective project planning.
- Effective project cost estimating.
- Effective project measurements.
- Effective project milestone tracking.
- Effective project quality control.
- Effective project change management.
- Effective development processes.
- Effective communications.
- Capable project managers.
- Capable technical personnel.
- Significant use of specialists.
- Substantial volume of reusable materials.

Failing Projects

- Inadequate project planning.
- Inadequate project cost estimating.
- Inadequate project measurements.
- Inadequate project milestone tracking.
- Inadequate project quality control.
- Ineffective project change management.
- Ineffective development processes.
- Ineffective communications.
- Ineffective project managers.
- Inexperienced technical personnel.
- Generalists rather than specialists.
- Little or no reuse of technical material.

Using the Spiral Model and MBASE to Generate New Acquisition Process Models: SAIV, CAIV, and SCQAIV

Dr. Barry Boehm, Dr. Dan Port, LiGuo Huang, and Winsor Brown
University of Southern California

In this article, we show how you can use the MBASE process framework to generate a family of acquisition process models for delivering user-satisfactory systems under schedule, cost, and quality constraints. We present the six major steps of the Schedule/Cost/Schedule-Cost-Quality as Independent Variable (SAIV/CAIV/SCQAIV) process using SAIV and a representative Department of Defense (DoD) Command, Control, and Communications Interoperability application as context. We then summarize our experience in using SAIV on 26 University of Southern California electronic services projects, followed by discussions of SAIV/CAIV/SCQAIV application in the commercial and defense sectors, of model application within the DoD acquisition framework, and of the resulting conclusions.

A number of Department of Defense (DoD) organizations are responding to the DoD Evolutionary Acquisition Initiative in DoDI 5000.2 [1] by organizing evolutionary increments of capability around the objective of developing and fielding each increment within a fixed schedule (frequently 18 or 24 months) or fixed budget. Examples are new capabilities or major upgrades for such software-intensive systems as Command, Control, and Communications Interoperability (C3I), logistics, or combat platform electronics suites.

The usual approach for achieving this objective follows this pattern:

1. Determine the best-possible set of features that can be developed and fielded within the available schedule and/or budget.
2. Contract to develop and field this feature set within the available schedule and/or budget.
3. Monitor the contractor's progress in achieving the objectives within the schedule and/or budget.

This is the usual interpretation of DoD's current Cost as Independent Variable (CAIV) approach. Unfortunately, step four of this scenario usually involves finding that the available schedule and/or budget are insufficient, and that the existing contract constraints and architectural commitments preclude finding a way to field an acceptable capability within the available schedule and/or budget.

Is this really the usual outcome? Sadly, yes, both in government and commercial software acquisition. For example, the Standish Report [2] found that 84 percent of the software-intensive system projects it surveyed either overran their budgets and schedules or were cancelled before completion. The average overruns on these projects were 189 percent of planned cost and 222 percent of planned schedule. The completed overrun projects delivered an average of only 61 percent of the originally specified features. The Standish Report does not address the effect on delivered software quality, but our analysis of similar projects indicates similar problems with delivered defect density (nontrivial defects per function point or per thousands of source lines of code).

> *"... 84 percent of the software-intensive system projects it [Standish] surveyed either overran their budgets and schedules or were cancelled ..."*

The Standish Report's and our analyses of the major root causes of this problem are:

- Schedule and budget estimates can be (sometimes wildly) optimistic.
- Even if "most likely" estimates are used, the definition of most likely means that they will be overrun in roughly half of the projects.
- To maximize the probability of successful delivery, the contractor will often use a point-solution architecture to accommodate the specified features. When the inevitable threat, combat platform, feature priority, or technology changes come, they are hard to accommodate within the point-solution architecture.

Using SAIV, CAIV, and SCQAIV as Process Models

In our earlier CrossTalk articles on the Spiral Model [3] and Model-Based (System) Architecting and Software Engineering (MBASE) [4], we showed that these were actually process model generators for the acquisition of software intensive systems. They use risk considerations to determine the most appropriate sequence of activities to perform (among specification, prototyping, simulation, benchmarking, increments of development, etc.) in order to achieve the most cost-effective system capability within various resource constraints such as cost, schedule, personnel, and platform characteristics.

In this article, we show how you can use the MBASE process framework to generate a particularly attractive family of acquisition process models for delivering user-satisfactory systems under schedule, cost, and quality constraints.

The risk-driven MBASE-Spiral approach uses the risk of schedule or cost overrun to invert the usual software-intensive-system acquisition process. Either schedule, cost, or some combination of schedule, cost, and quality becomes the independent variable, and the lower-priority features become the dependent variable. This requires several sub-processes:

- Determination of a top-priority core capability and quality level strongly assured to be achievable within the schedule-cost-quality constraints.
- User expectations management and continuing update of feature priorities.
- Architecting the system for ease of dropping borderline-priority features and future addition of lower-priority features.
- Careful progress monitoring and corrective action to keep within cost-schedule-quality constraints.

In this article, we next present the six major steps of the Schedule/Cost/Schedule-Cost-Quality as Independent Variable (SAIV/CAIV/SCQAIV) process using SAIV and a representative DoD C3I application as context. We then summarize our experience in using SAIV on 26 University of Southern California (USC) electronic services projects, 24 of which have successfully delivered systems with high client-satisfaction ratings on a fixed schedule. This is followed by discussions of SAIV/CAIV/SCQAIV application in the commercial and defense sectors, of model limitations and extensions, and of the resulting conclusions.

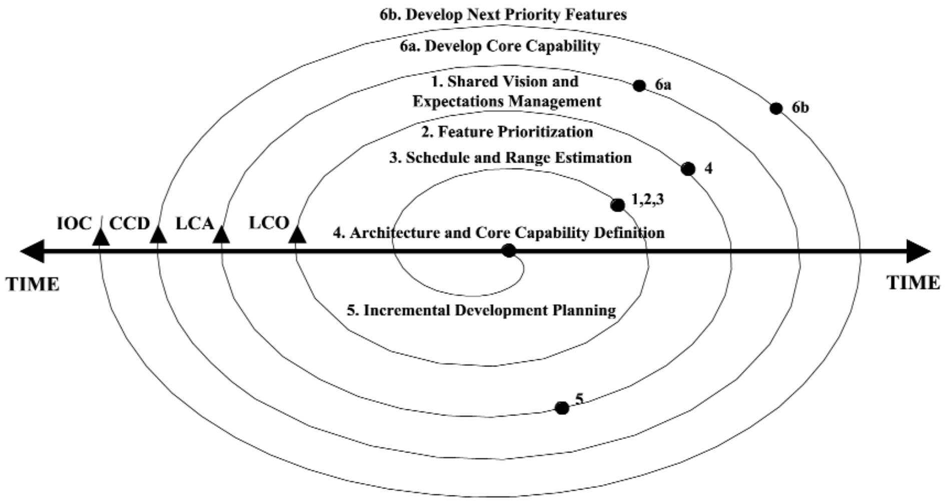

6b. Develop Next Priority Features
6a. Develop Core Capability
1. Shared Vision and Expectations Management
2. Feature Prioritization
3. Schedule and Range Estimation
4. Architecture and Core Capability Definition
5. Incremental Development Planning
6a
6b
4
1,2,3
5
IOC
CCD
LCA
LCO
TIME
TIME

Figure 1: *Mapping of SAIV Spiral Process Elements Onto Win-Win Model*

The SAIV Process Model

The key to successful SAIV practice is to strategically plan through all life-cycle areas to meet a delivery date. SAIV is defined by explicitly enacting the following six process elements:

1. Manage expectations by establishing a stakeholders' shared vision of achievable objectives.
2. Prioritize system features.
3. Estimate subsets of features that can be developed with high confidence within the available schedule.
4. Establish a coherent set of core capabilities with borderline features to be added if possible, and a software/system architecture to easily accommodate borderline features.
5. Plan development increments, including a high-confidence core capability and next-priority subsets.
6. Execute development plans with careful change and progress monitoring and control processes.

The MBASE process model generator is used to generate a SAIV process model suitable for a particular project. Figure 1 shows the SAIV version of the Win-Win Spiral Model. The process models for CAIV and SCQAIV are essentially the same except for the definition of the radial dimension of the spirals. For CAIV projects, the spiral's traditional radial dimension of cumulative cost is used. For SAIV projects, the radial dimension is cumulative calendar time. Either cost or time can be used for SCQAIV, with the other objective and desired quality acting as constraints.

Figure 1 also shows the major SAIV process elements to be described next. These are executed concurrently within the spirals. As discussed in our updated spiral model article [3], feedback and iteration of previous-cycle results are part of the spiral process, but are omitted from Figure 1 for simplicity.

The milestone content and pass-fail criteria for the Life Cycle Objectives (LCO), Life Cycle Architecture (LCA), and Initial Operational Capability (IOC) in Figure 1 were described in detail in December CROSSTALK's article on the Spiral Model and MBASE [4]. They are also the major development milestones in the Rational Unified Process [5, 6]. We will elaborate the LCA milestone content in a following section. The SAIV/CAIV/SCQAIV family of process models adds a further milestone in Figure 1: the Core Capability Demonstration (CCD). It will be detailed, too, in later sections.

A Representative C3I System

We now elaborate and illustrate the six SAIV steps in the context of a representative C3I system. The current system has three major upgrade requirements: changing to a Web-based operation; changing to an XML-based interoperability scheme; and adding a new weather-impact capability to support better operational planning, task planning, and battle management decision making. A new fielded capability is needed in 19 months to maintain compatibility with other interoperating systems transitioning to the Web and XML at that time.

Shared Vision and Expectations Management

As graphically described in Death March [7], many software projects lose the opportunity to assure a rapid, on-time delivery by inflating client expectations and over promising on delivered capabilities. The first step in the SAIV process model is to avoid this by obtaining stakeholder agreement that meeting a fixed schedule for delivering the system's IOC is the most critical objective, and that the other objectives such as the IOC feature content can be variable, subject to meeting acceptable levels of quality and post-IOC scalability.

For the example C3I system, the 19-month IOC milestone is clearly critical for interoperability. Early meetings of the system's integrated product team should emphasize that meeting this milestone may be incompatible with stakeholders getting all the features they want.

Feature Prioritization

With MBASE at USC, stakeholders use the USC/GroupSystems.com EasyWinWin requirements negotiation tool [8] to converge on a mutually satisfactory (win-win) set of project requirements. One step in this process involves the stakeholders prioritizing the requirements by assessing their relative importance and difficulty, each on a scale of zero to 10. This process is carried out in parallel with initial system prototyping, which helps ensure that the priority assessments are realistic.

Easy WinWin has been used successfully for DoD software applications [9]. However, other collaboration tools or even manual group-meeting techniques can be used for this step. In our C3I example, the stakeholders rate the Web and XML capabilities higher-priority based on interoperability essentials, but agree that Weather capabilities are important also.

Schedule Range Estimation

The developers then use a mix of expert judgement and parametric cost modeling to determine how many of the top-priority features can be developed in 24 weeks under optimistic and pessimistic assumptions. For the parametric model, we use Constructive Cost Model (COCOMO) II, which estimates 90 percent confidence limits on both cost and schedule [10]. Other models such as Software Life-Cycle Model (SLIM) [11], System Evaluation and Estimation of Resources (SEER) [12], and Knowledge PLAN [13] provide simi-

Capabilities	Fastest Achievable Schedule (months)	
	Most Likely (50% confidence)	90% confidence
Either Web or XML	12	15
Both Web and XML	15	19
Web, XML, and Weather	19	24

Table 1: *Fastest Achievable Schedules for C3I Capabilities*

lar capabilities.

Table 1 summarizes the results of a COCOMO II analysis of the example C3I system. It shows the fastest achievable schedules for completing either the Web or XML capabilities (each require 12 months at best); both the Web and XML; or all three capabilities (Weather requires 14 months at best). The two columns show the most likely schedule (achievable 50 percent of the time) and the 90-percent confidence schedule (achievable 90 percent of the time).

The stakeholders see that all three capabilities can be achieved in 19 months in the most likely estimate, but are concerned that this means that the 19-month schedule will be overrun about half the time; furthermore, that with 90 percent confidence it will take up to 24 months, an unacceptable outcome. However, the Web and XML capabilities could be completed in 19 months 90 percent of the time.

Architecture and Core Capability Determination

The most serious mistake a project can make at this point is just to pick the topmost priority features with 90 percent confidence of being developed in 19 months. This can cause two main problems: producing an IOC with an incoherent and incompatible set of features, and delivering these without an underlying architecture supporting easy scalability up to the full feature set and workload.

First, the core capability must be selected so that its features add up to a coherent and workable end-to-end operational capability. Second, the remainder of the lower-priority IOC requirements and subsequent evolution requirements must be used in determining a system architecture facilitating evolution to full operational capability. Still the best approach for achieving this is to use the Parnas information-hiding approach to encapsulate the foreseeable sources of change within modules [14]. The architecting process may take two or more win-win spiral cycles of prototyping, commercial off-the-shelf (COTS) product evaluation, and stakeholder renegotiation to reconcile the system's product, process, property, and success models into a LCA package.

The C3I system stakeholders determine that the core capability should include the critical subsets of the Web, XML, and Weather capabilities, rather than all of the Web and XML capabilities. This is both because the Weather decision support is much needed, and because it would be infeasible to add a significant Weather capability in just the time left after the core capability was completed.

Incremental Development Planning

The LCA package includes an incremental development plan (item 5 in Figure 1, page 21) indicating the schedules and pass/fail criteria for the core capability (item 6a), IOC (item 6b), and perhaps other milestones.

Since the core capability has only a 90 percent assurance of being completed in 19 months, this means that about 10 percent of the time, the project will have to stretch to deliver the core capabilities in 19 months, perhaps with some performer overtime or completion bonuses, or occasionally by further reducing the top-priority feature set. In the most likely case, however, the project will achieve its core capability with about 20 percent to 30 percent of the schedule remaining. This time can then be used to add the next-highest priority features into the IOC (again, assuming that the system has been architected to facilitate this).

An important step at this point is to provide the operational stakeholders (users, operators, maintainers) with a core capability demonstration. Often, this is the first point at which the realities of actually taking delivery of and living with the new system hit home, and their priorities for the remaining capabilities may change.

Also, this is an excellent point for the stakeholders to reconfirm the likely final IOC content, and to synchronize plans for conversion, training, installation and cutover from current operations to the new IOC.

Development Execution; Change and Progress Monitoring and Control

As progress is being monitored with respect to plans, there are three major sources of change that may require reevaluation and modification of the project's plans:

1. Schedule slips. Traditionally, these can happen because of unforeseen technical difficulties, staffing difficulties, customer or supplier delays, etc.
2. Requirements changes. These may include changes in priorities, changes in current requirements, or needs for new high-priority requirements.
3. Project changes. These may include staffing changes, COTS changes, or new marketing-related tasks (e.g., interim sponsor demos).

In some cases, these changes can be accommodated within the existing plans. If not, there is a need to rapidly renegotiate and restructure the plans. If this involves the addition of new tasks on the project's critical path, some other tasks on the critical path must be reduced or eliminated. There are several options for doing this, including dropping or deferring lower-priority features, reusing existing software, or adding expert personnel. In no cases should new critical-path tasks be added without adjustments in the delivery schedule or other schedule drivers.

By following these guidelines, the C3I project should be able to overcome the usual sources of change above and successfully deliver a core capability within 19 months, often with most of the full set of capabilities added as well. However, although SAIV can significantly improve your success rate, it can't guarantee success in all situations, such as major budget cuts or radical project redirections. In these cases, budget, schedule, and core capability content will need to be significantly renegotiated. Some examples of failed SAIV projects are given in the next section.

SAIV Experience

The SAIV process model is described in terms of a representative set of SAIV applications: USC's annual series of real-client campus e-services projects [15, 16]. These projects are largely Web-based applications developed by five-person master's-student teams, using the MBASE guidelines [17] and the MBASE Electronic Process Guide [18].

The teams' main challenges are to develop a LCO package and a LCA package, described below, for a USC

Information Services Division client's application in 12 weeks during the fall semester, and to develop and transition an IOC in 12 weeks during the spring semester. These are extreme examples of schedule being the independent variable since the USC semester schedule is fixed and the students disappear (to graduation or summer jobs) at the end of the spring semester.

The critical success factors of the MBASE process model involve the concurrent development of several initial artifacts: an operational concept description, a requirements definition, an architecture description, a life-cycle plan, a feasibility rationale, and one or more prototypes. These are evaluated at two major pass/fail points, the LCO and the LCA milestones. Both milestones use the same primary pass-fail criterion: If we build the system to the given architecture, it will satisfy the requirements, support the operational concept, be faithful to the prototypes, and be buildable within the processes, budgets, and schedules in the plan.

For the LCO milestone, this criterion must be satisfied for at least one choice of architecture, along with demonstration of a viable business case for the system and the expressed concurrence of all the success-critical stakeholders. For the LCA milestone, the pass-fail criterion must be satisfied for the specific choice of architecture and COTS components to be used for the system, along with continued business case viability and stakeholder concurrence, plus elimination of all major project risks or coverage of the risks in a risk management plan.

One of our primary goals in the project course is to give the students experience in risk management [19]. Our risk management lectures and homework exercises emphasize a list of the 10 most serious risk items: Personnel risks are number one, and budget-schedule risks are number two. The student projects' risk management plans must show how their team will avoid the risks of delivering an unsatisfactory LCA package in the first 12 weeks (fall semester), and of unsatisfactorily delivering and transitioning an IOC in the second 12 weeks (spring semester). The MBASE guidelines recommend that they adopt the SAIV model described in an earlier section; so far, all the projects have done this.

Also, we work in advance with the USC e-services clients to sensitize them to the risks of over-specifying their set of desired IOC features, and to emphasize the importance of prioritizing their desired capabilities. This generally leads to a highly collaborative win-win negotiation of prioritized capabilities, and subsequently to a mutually satisfactory core capability to be developed as a low-risk minimal IOC.

The projects' monitoring and control activities include the following:

- Development of a top-N project risk item list that is reviewed and updated weekly to track progress in managing risks (N is usually between five and 10).
- Inclusion of the top-N risk item list in the project's weekly status report.
- Management and technical reviews at several key milestones.
- Client reviews at other client-critical milestones such as the core capability demonstration.

The use of SAIV and these monitoring and control practices have led to on-time, client-satisfactory delivery and transition of 24 of the 26 products developed to date. One of the two failures was in our first year, when we tried to satisfy three clients by merging their image archive applications into a single project; we underestimated the complexity of the merge. As a result, "merging multiple applications" has become one of the major sources of project risk that we consider.

The second failure happened recently when a project that appeared to be on track at its transition readiness review, simply did not implement its transition plan when its client suddenly had to go out of town. We were not aware of this until the client returned after the semester was over and the students had disappeared to graduation and summer jobs. We have since revised our system of closeout reviews to eliminate this "blind spot" and related problem sources.

On the other 24 projects, client evaluations have been uniformly quite positive, averaging about 4.4 on a scale of one to five. A particularly frequent client evaluation comment has been their pleasure in being able to synchronize product transition on a specific fixed date with their other transition activities. Another pleasant surprise was the effect on clients' review timeliness: "You mean if I evaluate the prototype right away, I'll get more features in my IOC?" The e-services project artifacts can be reviewed on the class Web page <http://sunset.usc.edu/classes>.

E-Commerce Projects

One of our industrial affiliates, C-Bridge, Inc., uses a very similar SAIV process model, which enables them to consistently deliver e-commerce systems on fixed schedules between 13 and 26 weeks. Their Rapid Value approach uses milestones very similar to MBASE's LCO, LCA, and IOC milestones; their counterpart phases are named define, design, develop, and deploy. They use similar approaches in working in advance with their clients to ensure a workable SAIV scope and schedule, and in anticipating and pre-working potential transition problems to client-based operations and maintenance [20].

The CAIV and SCQAIV Process Models

Simply substituting "cost" for "schedule" in the SAIV process model described above provides you with an equally effective way to use CAIV as a process model. The SCQAIV model is a straightforward extension of CAIV and SAIV. It involves setting the system's quality goals (e.g., a delivered defect density of 0.3 nontrivial defects per thousand source lines of code (KSLOC), or of 0.03 nontrivial defects per function point), and tracking progress with respect to achieving the desired combination of schedule, cost, or quality goals.

If any of these goals becomes unachievable in delivering the current feature set, the project must drop enough lower-priority features to make the combination of goals achievable. There may be limits to the project's ability to do this, such as insufficient schedule to develop even a viable core capability. We have discussed this situation via a production-function perspective in [21].

DoD Acquisition Framework

In situations such as post-deployment upgrades and pre-planned product improvements, DoD can and often has implemented versions of SAIV/CAIV/SCQAIV as smoothly as they are done commercially. Frequently, in such situations, the organization's software maintenance budget and release cycle are relatively fixed. The biggest risk is to promise too much within these constraints, leaving the sacrifice of quality as the only way to meet budget and schedule. This inevitably leads to degradations of the software's maintainability, operational fitness, and future maintenance productivity.

Thus, a form of SCQAIV is the best option for software maintenance in which quality standards are set, infrastructure upgrades are given appropriate priorities, and lower-priority features are shed to meet cost, schedule, and quality objectives.

This approach is workable because

DoD's operations and maintenance acquisition practices are similar to their commercial counterparts. Budgets are generally not tied to premature promises of delivered features, and there is usually a long-term customer-supplier relationship with a shared product vision among the customer, supplier, and users. This continuing relationship usually increases mutual trust, the ability to share and respect each other's win conditions, and the ability to negotiate and, when necessary, readily adjust mutually satisfactory or win-win agreements and priorities.

Competitive Development

In some cases, such as the Air Force Electronic Systems Center's Command Center Product Line (CCPL) and within classified-application organizations, DoD customers have been able to create development arrangements similar to the stable post-deployment support situation described above. CCPL, for example, developed a flexible contractual instrument focused on creating user value rather than pre-specified features, and allowing in-process renegotiation of priorities. It selected three contractors via competitive source selection. The evaluation criteria included track record on similar developments, software Capability Maturity Model® (CMM®) process maturity, technical and management approach, and demonstration of the approach via a representative exercise.

Once selected, the contractors operated as a team with the customer, developing a strong shared vision for the product line, and taking on new assignments based on best-matched available expertise, ensuring effective employment of all three contractors' resources. In this situation, SAIV/CAIV/SCQAIV-type approaches were highly feasible and preferable.

In many cases, however, DoD organizations must develop a new system vision and set of acquisition parameters (schedule, cost, quality attribute levels, and feature scope) within a competitive acquisition framework. Here, complete multi-contractor shared vision development is impractical, as developers will be unwilling to share their competitive-discriminator technology solutions with competing developers.

Frequently, this leads acquisition organizations to exclude developers from participating in the creation of the shared vision. This is highly risky, as the resulting decision may exclude attractive developer technology solutions. It may also leave serious vision mismatches between the customer, user, and selected developers, making SAIV/CAIV/SCQAIV system scoping and feature prioritization difficult to achieve, particularly if the program's funding and schedule have been tied to a particular set of delivered capabilities.

Unfortunately, there is no ideal solution to this dilemma. The most attractive near-solution involves the use of multiple competitive spiral cycles of system definition, with the number of competitors being reduced from one cycle to the next. The earlier cycles are shorter and less expensive, making a larger number of participants affordable. They can be run as SAIV/CAIV procurements with equal opportunity for each competitor. Some care is necessary to avoid leaking competitors' key discriminators, but many competitive DoD concept definition efforts have achieved this.

> *"The six-step SAIV process model presented here has been used successfully on 24 of 26 e-service applications at USC ..."*

These earlier cycles enable overall system scoping and tradeoff analysis to be performed, along with the evaluation of readiness levels of key technologies via prototyping, benchmarking, modeling and simulation, etc. This also enables the acquirers to evaluate the competing developers' capabilities and understanding of the system context and objectives.

Some similar criteria to those used by CCPL (track record, technical and management capabilities, concept definition and evaluation performance) are used for initial competitor selection and early down-selection. Later down-selection criteria increasingly involve development capabilities such as process maturity and realism of development plans, schedules, and budgets. Here again, the final development competition can fix the cost and/or schedule, provide a prioritized feature set, and compete on scope and realism of feature set delivery plans.

All three services are making progress toward mastering this kind of evolutionary acquisition in the context of the new DoD 5000-series of acquisition regulations [1]. These include the extensive use of simulation and modeling in the Army's SMART program, the Air Force's Instruction 63-123, evolutionary acquisition of command and control systems [22], use of downselected contractors' expertise in other system life-cycle roles, and service use of new contractual vehicles such as cooperative research and development agreements (CRADAs) and the Defense Advanced Research Projects Agency (DARPA)-originated "other transactions" approach [22]. All of these are compatible with and have been used successfully with SAIV/CAIV/SCQAIV-type approaches.

Conclusions

The six-step SAIV process model presented here has been used successfully on 24 of 26 e-services applications at USC, and on a similar percentage of e-commerce applications at C-Bridge, to deliver highly client-satisfactory applications on a fixed schedule in a climate of rapid change. Its 92 percent success rate compares favorably with the 16 percent success rate in the Standish Group's survey of current practice. There might be some concern that student-team projects are easier than professional-developer projects, but there is also a case that on-schedule delivery is harder with teams of people who are unfamiliar with each other, with project practice, with the clients, and with the applications domain (all generally true for the USC e-services projects).

The critical success factors of the SAIV approach are:

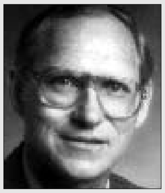
- Working with stakeholders in advance to achieve a shared product vision and realistic expectations.
- Getting clients to develop and maintain prioritized requirements.
- Scoping the core capability to fit within the high-payoff segment of the application's production function for the given schedule.
- Architecting the system for ease of adding and dropping borderline features.
- Disciplined progress monitoring and corrective action to counter schedule threats.

The approach can also be applied to its counterpart CAIV process model. It can also provide a way to transform the current dilemma, "Schedule, Cost, Quality: Pick Any Two," to "Schedule, Cost, Quality: Pick All Three," via the SCQAIV version of the model whenever your project is able to shed lower-priority features to meet its SCQ objectives. Proven strategies are also available for applying SAIV/CAIV/SCQAIV to competitive DOD system acquisitions.◆

References

1. Department of Defense Instruction 5000.2. Operation of the Defense Acquisition System. Sept. 2000, <www.acq.ord.mil/ap/i50002p.doc>.
2. Standish Group. "CHAOS," <www.standishgroup.com/chaos.htm>.
3. Boehm, B., and W. Hansen. "The Spiral Model as a Tool for Evolutionary Acquisition," CROSSTALK, May 2001: pp. 3-11.
4. Boehm, B., and D. Port. "Balancing Discipline and Flexibility with the Spiral Model and MBASE," CROSSTALK, Dec. 2001.
5. Kruchten, P. The Rational Unified Process. Addison-Wesley, 1998.
6. Royce, W.E. Software Project Management: A Unified Framework. Addison-Wesley, 1998.
7. Yourdon, E. Death March. Prentice Hall, 1997.
8. Boehm, B., P. Gruenbacher, and R. Briggs, eds. "Developing Groupware for Requirements Negotiation: Lessons Learned," IEEE Software. May/June 2001: pp. 46-55.
9. Saboe, M., P. Gruendacher, and P. Kloska. "Experience with the WinWin Process for Planning Software Infrastructure and Technology," Proceedings, International Conference of Good Systems, 2002 (to appear).
10. Boehm, B., C. Abts, A.W. Brown, S. Chulani, B.Clark, E. Horowitz, R.Madachy, D. Reifer, and B. Steece, eds. Software Cost Estimation with COCOMO II. Prentice Hall, 2000.
11. Putnam, L. "Software Life Cycle Model (SLIM)," QSM, 2001, <www.qsm.com>.
12. Galorath, D. SEER-SEM. Galorath, Inc., 2001,<www.galorath.com>.
13. Jones, C. "Knowledge PLAN," Artemis/SPR, 2001, <www.spr.com>.
14. Parnas, D., "Designing Software for Ease of Extension and Contraction," IEEE Trans. Software Engr. Mar. 1979: pp. 128-137.
15. Boehm, B., A. Egyed, J. Kwan, D. Port, A. Shah, and R. Madachy, eds. "Using the WinWin Spiral Model: A Case Study." IEEE Computer, July 1998: pp. 33-44.
16. Boehm, B., A. Egyed, D. Port, A. Shah, J. Kwan, and R. Madachy, eds. "A Stakeholder Win-Win Approach to Software Engineering Education," Annals of Software Engineering, Apr. 1999.
17. Boehm, B., D. Port, M. Abi-Antoun, and A. Egyed, eds. Guidelines for Model-Based Architecting and Software Engineering (MBASE). Ver. 2.2. USC-CSE, Feb. 2001, <http://suset.usc.edu/Research/MBASE>.
18. Mehta, N., MBASE Electronic Process Guide. USC-CSE, Sept. 1999, <http://sunset.usc.edu/Research/MBASE>.
19. Port, D. and B. Boehm, eds. "Educating Software Engineering Students to Manage Risk," Proceedings, ICSE 2001, May 2001.
20. Madachy, R., A. Pan, and A. Arboleda, eds. "Processes for Rapid Development of Internet Applications," LA SPIN Presentation, 24 Jan. 2001.
21. Boehm, B., and A.W. Brown, eds. "Mastering Rapid Delivery and Change with the SAIV Process Model," Proceedings, ESCOM2001, Apr. 2001.
22. U.S. Air Force Instruction 63-123, "Evolutionary Acquisition for C2 Systems," Apr. 2000, <http://afpubs.hq.af.mil>.

About the Authors



Barry Boehm, Ph.D., is the TRW professor of software engineering and director of the Center for Software Engineering at the University of Southern California. He was previously in technical and management positions at General Dynamics, Rand Corp., TRW, the Defense Advanced Research Projects Agency, where he managed the acquisition of more than $1 billion worth of advanced information technology systems. Dr. Boehm originated the spiral model, the Constructive Cost Model, and the stakeholder win-win approach to software management and requirements negotiation.

University of Southern California
Center for Software Engineering
Los Angeles, CA 90089-0781
Phone: (213) 740-8163
Fax: (213) 740-4927
E-mail: boehm@sunset.usc.edu

Daniel Port, Ph.D., is a research assistant professor of Computer Science and an associate of the Center for Software Engineering at the University of Southern California. He received a doctorate degree from the Massachusetts Institute of Technology, and a bachelor's degree from the University of California, Los Angeles. His previous positions were assistant professor of Computer Science at Columbia University, director of Technology at the USC Annenburg Center EC2 Technology Incubator, co-founder of Tech Tactics, Inc., and a project lead and technology trainer for NeXT Computers, Inc.

University of Southern California
Center for Software Engineering
Los Angeles, CA 90089-0781
Phone: (213) 740-7275
Fax: (213) 740-4927
E-mail: dport@sunset.usc.edu

LiGuo Huang is a Ph.D. student and research assistant in the computer science department at the Center for Software Engineering at the University of Southern California (USC). She received a bachelor's degree in engineering management from HoHai University, China, and a master's degree in computer science from USC. Her current research interests focus on software system's Schedule/ Cost/Schedule-Cost-Quality as Independent Variable process model and economic-driven software engineering. She was an Honored Student of JiangSu Province awarded by the Education Committee of JiangSu Province, China, an Honored Student of Hohai University, and was awarded the Outstanding Academic Achievement Award at USC.

University of Southern California
Computer Science Department
Los Angeles, CA 90089
Phone: (213) 740-6505
Email: liguohua@usc.edu

Project Recovery ... It Can be Done

Walt Lipke
Oklahoma City Air Logistics Center

There are courses of action available to project managers when projects go awry. However, when the management indicators clearly point to project performance problems, their connection to the appropriate action is not easily determined. What action should the project manager take, and to what extent? How can the manager know if the action taken will achieve the desired result? This paper addresses these questions.

Earned value management (EVM) provides project managers with a considerable amount of information concerning the health of their project's performance [1]. The project manager has detailed knowledge of the project's performance baseline, and several choices for assessing the performance status. Among the most common management methods using EVM data is the evaluation of cost variance (CV): the budgeted cost for work performed (BCWP) minus the actual cost for work performed (ACWP); and schedule variance (SV): the BCWP minus the budgeted cost for work scheduled (BCWS). The project manager normally will establish "triggers," or percentages for the variances, which, if exceeded, cause a project review and possible management action.

Within the Oklahoma City Air Logistics Center's (OCALC) Software Division, we prefer using the efficiency indicators of EVM to gauge project status [2]. Of course, these indicators are the cost and schedule performance indexes, CPI and SPI, respectively. CPI is the ratio of BCWP to ACWP, while SPI is the ratio of BCWP to BCWS. Fundamentally, the same information is available from these indicators and the variances, CV and SV.

Regardless of the indicators chosen, the information can be viewed as totals, commonly termed "cumulative," or by time periods such as monthly. The cumulative indicators represent all cost and schedule progress from the project start until the present. The monthly indicators provide information for a specific month, and are very useful for recognizing performance trends, good or bad.

Thus, project managers have information to assist their efforts. What can they do with it? Certainly the objective of the project manager is to use the information to control their project and achieve its cost, schedule, and technical performance requirements. Following these fundamental needs are goals for customer satisfaction, company profit, and employee rewards. There is a lot at stake and a considerable amount of pressure to do the job well.

> *"... determining the management action ... is not simple; it is not all that easy to realign employees to maximize efficiency, or to remove inefficient workers."*

Project Control

When the project performance is not as good as expected, or needed, four management actions are possible:

1. The level of overtime or number of employees on the project can be raised or lowered.
2. Employees can be realigned to increase the efficiency of the project.
3. The performance requirements of the project can be reduced.
4. Additional funding and (or) schedule can be added to the project.

Normally, actions one and two are within the project manager's prerogative; he/she can adjust overtime, change the staffing, and realign personnel. Actions three and four, however, usually require negotiation. These actions involve higher management and components of the organization (e.g., contracting), which are beyond the project manager's control. Certainly, actions three and four are appropriate when the situation warrants, but they have the potential to be damaging to all concerned: company, customer, project manager, and the employees. Future business is at risk when customers are informed the product cannot be delivered at the original price and schedule. Re-negotiation with an agitated customer is not a pleasant experience.

The manager that can keep the project moving towards its objectives, thereby avoiding re-negotiation, is a successful one. Skillful employment of available staff and overtime are critical to a positive project outcome. These observations are intuitive. However, to effectively choose between the possible management actions and appropriately control staffing and overtime, the following questions require answers:

1. How does the project manager determine what type of recovery action is appropriate?
2. Once determined, what should be the extent of the action?
3. How does the project manager know if the determined action is achievable and is not an overreaction?

The remainder of this article responds to these questions. The majority of the discussion is focused on question three.

The Basics

As mentioned earlier, using the SPI and CPI cumulative values is the normal management practice within the OCALC Software Division. When the SPI and (or) CPI indicates poor performance, the pairing of the indicators leads to a management action [2]. If the action is to adjust overtime or number of employees, the equations provided in Figure 1 (see page 27) are then used. The result of the calculation is the staffing or overtime necessary to correct the performance for achieving the project plan.

Take note of the "Band the Recovery" words in Figure 1. The minimum action required to achieve customer needs occurs when all of the management reserve is consumed. The expectation is for management reserve to be totally consumed when the value of cost ratio (CR), or schedule ratio (SR), is used in the calculation[1] [2]. With the results of the two calculations, the project manager has the potential maximum and minimum responses to the problem; he/she can then select a performance correction approach between the extremes.

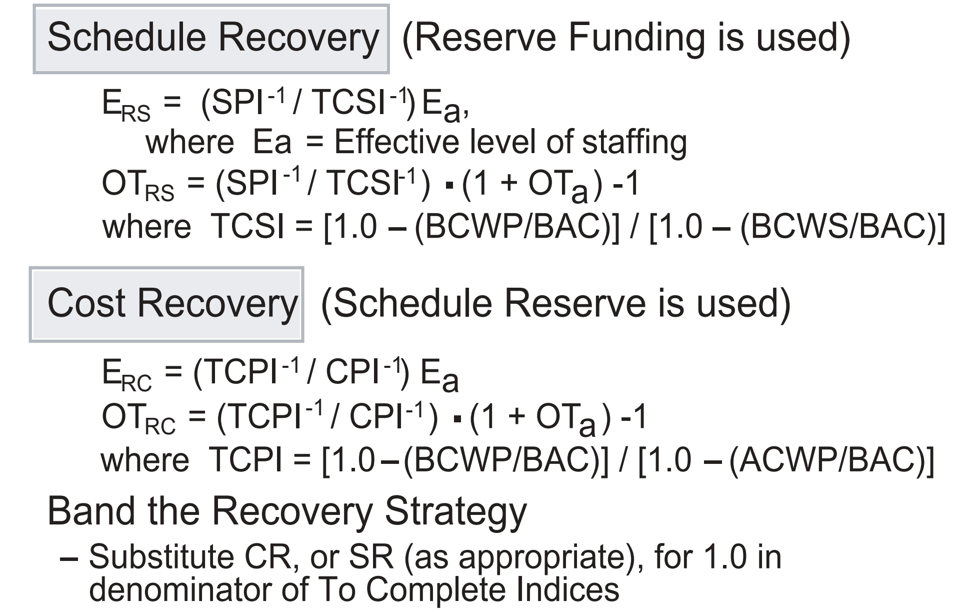
To this point, determining the management action and adjustments appears to be relatively simple. If we want to recover schedule, we must add staff or (and) increase overtime. If we want to improve cost performance, we must realign and (or) decrease staff. It may not be complex, but this is not "simple;" it is not all that easy to realign employees to maximize efficiency, or to remove inefficient workers. It is a tough situation that requires great inter-personal skills. These are the moments in a project when being a project manger is not much fun.

Also, once poor performance has been established there is very little chance of getting the project back on track with the performance baseline. It is certainly not easy to break this news to upper management, either. However, we must accept that the budget at completion (BAC), the planned completion date, and cost of the project, will be exceeded. And, unless a miracle occurs, i.e., the project achieves real performance efficiency improvements, our efforts to "control" the project will cause some consumption of the management reserve. Recovery is not free; schedule recovery will increase cost, and improving cost efficiency will lengthen schedule.

Refining the Strategy

The simple use of the SPI and CPI to adjust overtime and staffing described earlier is helpful, but we have not answered question three posed in the section on "Project Control." We don't know if the adjustment is an overreaction, or if it is achievable. By taking action without answering this question, the project manger could be setting the project up for additional problems instead of correcting its performance.

To answer the question, we need to know more. We must be able to determine if implementing a change (overtime, staffing) can correct performance to expected completion within the limits of its management reserve. Presently, managers calculate the estimate at completion (EAC) [1] as a check for cost, but they do not have a comparable calculation for testing the impact of a change to the schedule. As was discussed previously, a change in cost performance impacts schedule performance, and vice versa. However, although we understand there is a relationship between them, we do not have a description that models the behavior.



Schedule Recovery (Reserve Funding is used)

$E_{RS} = (SPI^{-1} / TCSI^{-1})\, E_a$,

where Ea = Effective level of staffing

$OT_{RS} = (SPI^{-1} / TCSI^{-1}) \cdot (1 + OT_a) - 1$

where TCSI = [1.0 – (BCWP/BAC)] / [1.0 – (BCWS/BAC)]

Cost Recovery (Schedule Reserve is used)

$E_{RC} = (TCPI^{-1} / CPI^{-1})\, E_a$

$OT_{RC} = (TCPI^{-1} / CPI^{-1}) \cdot (1 + OT_a) - 1$

where TCPI = [1.0 – (BCWP/BAC)] / [1.0 – (ACWP/BAC)]

Band the Recovery Strategy

– Substitute CR, or SR (as appropriate), for 1.0 in denominator of To Complete Indices

Figure 1: *Adjusting Overtime and Employees*[2]

The Model

This section requires some knowledge of calculus and differential equations. If the reader is unfamiliar with these areas of mathematics, he/she may skip to the end of the section. Near the end of the model section, the reader should review equations five and six before proceeding to the application section. The remainder of the article does not require understanding the derivation of the model.

To begin developing the model, we recognize that a change in cost performance, for example, induces a proportional "negative" change in the schedule performance (and vice versa). We alluded to this fundamental concept earlier in the article; the preceding sentence is, simply, a more mathematical way of stating the observation. Thus, in equation form

$$(1) \quad \Delta c \;\hat{}\; -\Delta s$$

where Δ symbolizes the change in performance with the subscript c denoting cost, and s schedule; the symbol ^ indicates "proportional to." Writing this equation in the terms of the performance indicators, it becomes

$$(CPIr^{-1} - CPI_a^{-1}) \;\hat{}\; - (SPI_r^{-1} - SPIa^{-1})$$

where the subscript r indicates the recovery value, and a is the current (actual) value.

If the project has been executing for a reasonable period of time, the cumulative values of CPI and SPI define a "state" of performance. It is the relationship of the project execution to its plan. The multiplication product of SPIa^{-1} and CPIa^{-1} represents that state of performance. When overtime or staffing changes are made, the state of performance tends to remain as it was; in general, there is inertia to any change. Stated mathematically,

$$(SPI_a^{-1} + \Delta s) * (CPI_a^{-1} + \Delta c) = SPI_a^{-1} * CPI_a^{-1}$$

where the symbol * indicates multiplication.

Assuming the Δ values are not large, then the following relationship is determined

$$(2) \quad (\Delta s / SPI_a^{-1}) \; + \; (\Delta c / CPI_a^{-1}) \; \geq 0$$

Using the equality and rearranging the terms we obtain

$$(3) \quad \Delta s / \Delta c \; = - \, SPI_a^{-1} / CPI_a^{-1}$$

Thus, after some algebraic manipulation, it can be deduced from equation three that the assumption made about the "state" of the performance yields our beginning observation, i.e., a change in cost performance induces a proportional negative change in schedule performance, and vice versa" (see equation 1). From the mathematics of calculus, equation (3) can be restated: the slope of the SPI^{-1} versus CPI^{-1} function evaluated at point a is equal to the negative of the ratio:

$$SPI_a^{-1}/CPI_a^{-1}$$

A significant point to understand regarding changing the performance characteristics of a project is that any change induces inefficiency. For example, if people are added to a project to improve schedule performance, even if they are wonderful employees and are skilled at doing the work, they will still require an orientation time to become familiar with their roles and interfaces. Instant performance improvement is not possible.

Also, as the necessary change in performance increases, the amount of inefficiency experienced from implementing the change commensurately increases. To effectively manage the performance change, we need to understand the "principle of diminishing returns." Sometimes, our best course of action is to do nothing with the project performance. For this case, the only option remaining is negotiation.

In previous discussion, we had stated that to recover CPI^{-1} or SPI^{-1} to its planned value of 1.0 is virtually impossible. Mathematically, a description of this relationship, when CPI^{-1} is the indicator to be improved, can be stated in calculus notation as

$$dSPI^{-1} / dCPI^{-1} = -\infty, \text{ as } CPI^{-1} \text{ approaches } 1.0$$

where the symbol ∞ means an infinitely large number, and $dSPI^{-1}/dCPI^{-1}$ is calculus notation for the first derivative of the function SPI^{-1} with respect to the variable CPI^{-1}. The calculus equation describing this relationship is

$$(4) \quad dSPI^{-1} / dCPI^{-1} = -k / (CPI^{-1} - 1.0) \text{ (see Note 3)}$$

where k is a constant whose value is determined by evaluating this equation at point a and equating the result to the calculus restatement of equation (3),

$$dSPI^{-1} / dCPI^{-1} = - SPI_a^{-1} / CPI_a^{-1}$$

The constant k is, thus, determined to be

$$k = (CPI_a^{-1} - 1.0) * (SPI_a^{-1} / CPIa^{-1})$$

where, again, the subscript a indicates the values of CPI^{-1} and SPI^{-1} are at point a.

After substituting this expression for k into equation (4), the differential equation is then solved, thereby providing the following result for cost performance recovery (i.e., when CPI^{-1} is poor and SPI^{-1} is satisfactory)

$$(5) \quad SPI_r^{-1} = SPI_a^{-1} + (CPI_a^{-1} - 1.0) (SPI_a^{-1} / CPI_a^{-1}) \ln [(CPI_a^{-1} - 1.0) / (CPI_r^{-1} - 1.0)]$$

where ln in the equation is the abbreviation for the natural logarithm, and the subscript r denotes the recovery value.

The equation for schedule performance recovery (when SPI^{-1} is poor and CPI^{-1} is satisfactory) can be analogously determined by simply interchanging CPI^{-1} and SPI^{-1} in the derivation of equation (5). The resultant equation for schedule recovery is

$$(6) \quad CPI_r^{-1} = CPI_a^{-1} + (SPI_a^{-1} - 1.0)] (CPIa^{-1} / SPI_a^{-1}) \ln [(SPI_a^{-1} - 1.0) / (SPI_r^{-1} - 1.0)]$$

Application

To illustrate the use of the recovery equations, we will apply them to an example. For the hypothetical project, we have the following data: CR = 1.2, SR = 1.3, (ACWP/BAC) = 0.5, (BCWP/BAC) = 0.4, (BCWS/BAC) = 0.45. Thus, $CPIa^{-1}$ = 1.250 and $SPIa^{-1}$ = 1.125. Comparing the performance indexes to their respective ratios, we can see that cost performance is poor ($CPIa^{-1}$ > CR). While schedule performance is not as planned ($SPIa^{-1}$ > 1.0); it is expected to meet the customer requirement ($SPIa^{-1}$ < SR). Therefore, our job as a project manager is to improve cost efficiency while retaining satisfactory schedule performance.

The use of equation (5) for determining a cost recovery strategy is not straightforward because the function is logarithmic. One fairly simple method is to make a few calculations for $CPIr^{-1}$ and $SPIr^{-1}$ using the equation, and then graph their coordinates using log-linear graphing paper. Having the graph of the function will then allow selection of a viable recovery value. A viable recovery value is determined when both CPI^{-1} and SPI^{-1} are less than their respective ratio, CR and SR. If no such coordinate can be found, recovery is impossible; negotiation is the only management option remaining.

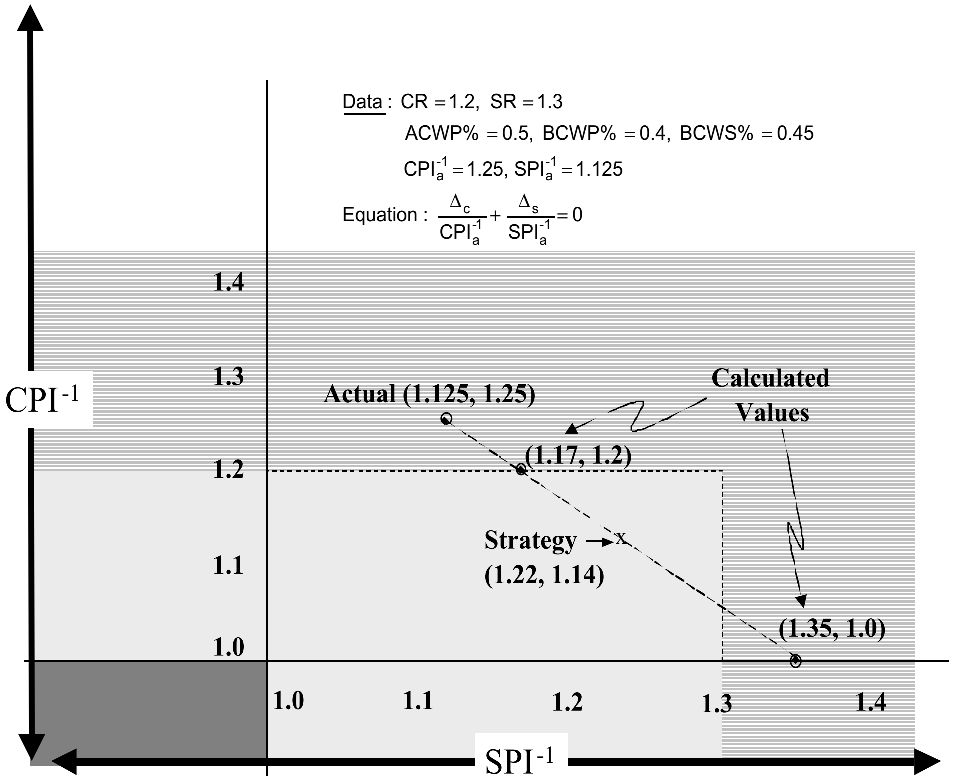
A much easier graphical approach for identifying viable recovery values is to approximate the SPI^{-1} and CPI^{-1} relationship by using equation (2), i.e., a straight line. For the simple method, two steps are needed. First, the straight-line approximation solution is obtained. A possible viable recovery coordinate is selected, and then it is tested using the logarithmic equation to provide assurance the solution is within the CR and SR values. For our example, solve equation (2) for $SPIr^{-1}$ (see Note 4) to obtain

$$SPIr^{-1} = SPIa^{-1} + (SPIa^{-1} / CPIa^{-1}) * (CPIa^{-1} - CPIr^{-1})$$

Next, substitute the actual values into the above equation and calculate values for $SPIr^{-1}$ corresponding to $CPIr^{-1}$ equal to CR (1.20) and 1.0. Using the results from these calculations along with the actual values a straight line can be plotted as shown by Figure 2.

The selected strategy of $SPIs^{-1}$ = 1.220 with $CPIs^{-1}$ = 1.140 is now tested

Figure 2: *Project Recovery Example*



by using the logarithmic equation. The strategy value of $CPIs^{-1} = 1.140$ is used in the equation to re-compute $SPIs^{-1}$. This computation yields a value of $SPIs^{-1} = 1.256$. Both the selected and computed values of the performance indexes are less than their respective ratios, thus we know the strategy is achievable and is not an overreaction.

Staffing and Overtime Adjustments

Now that we have a recovery strategy for the project, we'll illustrate how it can be used to adjust staffing and overtime. For our hypothetical project, assume we are presently staffed with 20 engineers who are working at 7 percent overtime. Essentially, we will adjust staffing and overtime by using the "To Complete" index corresponding to the recovery strategy.

For cost recovery, the To Complete Performance Index[5] (TCPI) is computed as follows

$$\textbf{TCPI} = [1.0 - (BCWP / BAC)] / [CPI_s^{-1} - (ACWP / BAC)] = [1.0 - 0.4] / [1.14 - 0.5] = 0.9375$$

(see Note 6)

Using the TCPI value, the staffing for the remainder of the project can be determined

$$Es = (TCPI^{-1} / CPI_a^{-1}) * Ea = (1.067 / 1.25) * 20 = 17.1 \text{ engineers @ } 7\% \text{ overtime}$$

where E is the number of engineers, and the subscripts s and a indicate the strategy and present values, respectively.

And, similarly overtime can be calculated

$$OTs = (TCPI^{-1} / CPI_a^{-1}) * (1 + OTa) - 1.0 = (1.067 / 1.25) * (1.07) - 1.0 = -0.9\% \text{ @ } 20 \text{ engineers}$$

where OT is the overtime rate. As you can plainly see, reducing overtime for this strategy is not an option; negative overtime is impossible. Therefore, staffing must decrease. We will now re-compute the overtime corresponding to the staffing of 17 engineers

$$OTs = (17.1 / 17) (1.07) - 1.0 = 7.6\%$$

Thus, the implementation of the recovery strategy is to reduce staffing by three software engineers and increase overtime by 0.6 percent.

Summary

This paper has provided the tools for constructing and implementing a project recovery strategy. A mathematical model of the recovery relationship between the EV indicators, SPI^{-1} and CPI^{-1}, was developed. An application of the relationship was discussed using a cost recovery example. The example illustrates how to obtain a recovery strategy, and then how to translate the strategy into personnel and overtime adjustments. Effective implementation of the adjustments should correct the project performance and result in the achievement of the customer requirements for cost and schedule. Effective, viable, project recovery can be accomplished.◆

References

1. Fleming, Quentin. Cost/Schedule Control Systems Criteria, The Management Guide to C/SCSC. Chicago: Probus 1988.
2. Lipke, Walter H. "Applying Management Reserve to Software Project Management." CrossTalk Mar. 1999.

Notes

1. The cost ratio (CR) is defined as CR = TFA / BAC, where TFA is the total funding available to the project, i.e., BAC plus the management reserve. The schedule ratio (SR) is defined as SR = NPoP / PPoP, where NPoP is the negotiated period of performance and PPoP is the planned period of performance in workdays.
2. The definitions of the abbreviations and subscripts used in Figure 1 are as follows:
   a = actual
   BAC = Budget at Completion
   CR = Cost Ratio
   E = number of employees
   OT = overtime
   RC = cost recovery
   RS = schedule recovery
   SR = Schedule Ratio
   TCPI = To Complete Performance Index
   TCSI = To Complete Schedule Index
3. Equation (4) is not the only mathematical form that becomes infinite as CPI^{-1} approaches 1.0. Because the mathematical relationship may be of another form, the computation results from using the derived equations (5) and (6) should be considered approximations. Other possible mathematical forms have been examined; only very small differences were seen in the computed values of $CPIr^{-1}$ and $SPIr^{-1}$.
4. If instead, schedule recovery were needed, equation 2 would be solved for $CPIr^{-1}$.
5. If the example had required schedule recovery instead of cost recovery, the equation for To Complete Schedule Index (TCSI) would have been used. The equation for TCSI is

$$\textbf{TCSI} = [1.0 - (BCWP / BAC)] / [SPI_s^{-1} - (BCWS / BAC)]$$

6. For recovery to be viable, the calculated value of TCPI (or TCSI, when recovering schedule) should be 1.0 or less. A TCPI greater than 1.0 indicates recovery performance must be better than the plan, which is not a reasonable expectation.

About the Author

Walt Lipke is the deputy chief of the Software Division at the Oklahoma City Air Logistics Center. The division employs approximately 600 people, primarily electronics engineers. He has 30 years of experience in the development, maintenance, and management of software for automated testing of avionics. In 1993 with Lipke's guidance, the Test Program Set (TPS) and Industrial Automation (IA) functions of the division became the first Air Force activity to achieve the Software Engineering Institute's Capability Maturity Model® (CMM®) Level 2. In 1996, these functions became the first software activity in federal service to achieve CMM Level 4 distinction. The TPS and IA functions, under Lipke's direction, became ISO 9001/TickIT registered in 1998. These same functions were honored in 1999 with the Institute of Electrical & Electronics Engineers Computer Society Award for Software Process Achievement. Lipke is a professional engineer with a master's degree in physics.

OC-ALC/LAS
Tinker AFB, OK 73145-9144
Phone: (405) 736-3341
Fax: (405) 736-3345
E-mail: walter.lipke@tinker.af.mil

CIO Update: The Expanding Responsibilities

L. John Michel
Information Resources Management College, National Defense University

It has been five years since the 1996 Clinger-Cohen Information Technology Management Reform Act was enacted mandating that all federal executive agencies have chief information officers (CIOs). Since then additional legislation and directives have expanded the role and responsibilities of federal CIOs. This article discusses how in today's world of e-commerce and e-government, federal CIOs must tackle infrastructure, architectures, information assurance, acquisition, software, information security, capital planning, human resource management, education, and other initiatives designed to improve the management of information technology resources.

The closing general session of the 1997 Software Technology Conference (STC) focused on the chief information officer (CIO) roles and responsibilities, and the 1966 Clinger-Cohen Information Technology Management Reform Act mandating that all federal executive agencies have CIOs. Attendees heard the Department of Defense (DoD) perspective from Cythnia Rand, who at that time was the principal director for Information Management, assistant secretary of defense for Command Control Communications and Intelligence (C3I). The CIO of Hewlett-Packard and the vice president of Strategic Business Initiatives from EDS gave industry's perspective.

The myth that preoccupies information systems executives is that CIO stands for "career is over [1]." Sitting in the ballroom of the Salt Place Convention Center in Salt Lake City during the STC conference, an attendee next to me questioned the need for the software community to understand the Clinger-Cohen act and the position of CIO. That person was oblivious to the transformation. Information intensive systems and the advances in information technology (IT) were fueling the revolution in military affairs within DoD and the revolution in business affairs in the federal government. As those revolutions continue today, CIOs' responsibilities are increasing with the passage of additional legislation and issuance of directives to meet the challenges of "e-government." CIOs are charged with a wide-ranging set of duties that provide the threaded connection needed to manage cyber space.

For that person who was sitting next to me, I hope that by the end of this article you see that in an ever-increasing information-intensive environment, CIOs have a crucial seat at the table. This article begins by reviewing the original act in order to set the foundation for analyzing the additional directives and legislation that have expanded the responsibilities of federal CIOs.

The Clinger-Cohen Act Revisited

The Clinger-Cohen Act, Division E of the fiscal year 1996 Defense Authorization Act, Public Law 104-106 (formerly the Information Technology Management Reform Act) was signed by President Clinton in 1996. It repealed the 1965 Brooks Act and directed federal agencies to put modern IT management frameworks into effect. The act established the position of the CIO for the executive agencies. In chartering this position, the framers of the act set forth certain responsibilities that CIOs should have.

> ***"CIOs' responsibilities are increasing with the passage of additional legislation and issuance of directives to meet the challenges of e-government."***

> "(1) Providing advice and other assistance to the head of the executive agency and other senior management personnel of the executive agency to ensure that IT is acquired and information resources are managed for the executive agency in a manner that implements the policies and procedures of this division, consistent with chapter 35 of title 44, United States Code, and the priorities established by the head of the executive agency;
> (2) Developing, maintaining, and facilitating the implementation of a sound and integrated IT architecture for the executive agency; and
> (3) Promoting the effective and efficient design and operation of all major information resources management processes for the executive agency, including improvements to work processes of the executive agency [2]."

Additionally, the act tasked the CIO with monitoring the performance of IT programs. Based on applicable performance measurements, the CIO would advise the head of the agency regarding whether to continue, modify, or terminate a program or project. The act stipulated that CIOs assess the requirements for agency personnel regarding knowledge and skill in information resources management, and develop strategies and specific plans for hiring, training, and professional development [3].

Following are other important components of the act to remember:

1. It repealed the requirement that agencies go through the General Services Administration for IT acquisitions.
2. Office of Management and Budget (OMB) has oversight control over agency IT spending through the budgeting process.
3. Agencies were required to create a process for maximizing the value and assessing and managing the risk of the IT acquisition.
4. Agencies had to develop performance measurements for IT that will measure how well the technology supports the programs of the agency.
5. Standards and guidelines are compulsory and binding to improve the efficiency of operation or security, and

privacy of federal computer systems.

6. Agencies acquire IT incrementally through the use of modular contracting.
7. Procurement protest authority now resides with the U.S. comptroller general and the General Accounting Office (GAO).

Two terms in the act are worth reviewing: IT and ITN National Security Systems (NSS). Clinger-Cohen defines IT in such a way that it encompasses business, command, control (C2), communications (C3), computer (C4), and intelligence (C4I) systems, and embedded systems:

> "…any equipment, or interconnected system or subsystem of equipment, that is used in the automatic acquisition, storage, manipulation, management, movement, control, display, switching, interchange, transmission, or reception of data or information by the executive agency. [It] includes computers, ancillary equipment, software, firmware, and similar procedures, services (including support services), and related resources [4]."

The Clinger-Cohen Act defines a NSS as "… any telecommunications or information system operated by the United States government, the function, operation, or use of which:

1. Involves intelligence activities;
2. Involves cryptologic activities related to national security;
3. Involves command and control of military forces;
4. Involves equipment that is an integral part of a weapon or weapons system; or
5. Is critical to the direct fulfillment of military or intelligence missions [5]."

Even though the act does not apply to NSS, there are numerous exceptions that bring these systems under the purview of the CIO. Section 5123 Performance- and Results-Based Management, Section 5126 Accountability, and Sections 5112 and 5122 Capital Planning and Investments Control all apply to NSS [6]. This was reinforced in a 1997 Office of the Secretary of Defense memorandum that stated, "Recent guidance from OMB places added emphasis on managing investments, to include weapons systems [7]."

For our discussion on the responsibilities of the CIO, it is important to remember that the major responsibilities, Section 5125, apply to the NSS. Thus the DoD CIO provides advice and assistance to the secretary of defense to ensure that NSS and IT resources are acquired consistent with law and policy. As we shall see this advice and assistance role has been incorporated into the acquisition life cycle.

Clinger-Cohen was a foundation for improving agency performance. Throughout the first years of implementation, CIOs worked to overcome the challenges faced with fulfilling new legislative requirements. Although Clinger-Cohen mandated enterprise architectures, it did not specify underlying components of standards and interoperability. In DoD, publication of Joint Vision (JV) 2010 established technological innovation as a key enabler and interoperability as the foundation; legislation was enacted that enables the DoD CIO to move from JV 2010 to JV 2020.

Expanding the Responsibilities of the DOD CIO

JV 2020 relies on IT and information systems to achieve its goal of full spectrum dominance through the operational concepts of dominant maneuver, precision engagement, focused logistics, and full dimensional protection. To achieve this dominance requires information superiority, joint C2 information operations, and a foundation of interoperability. Business applications (logistics, transportation, medical, and personnel) are certainly key enablers in JV 2020. Under the general responsibilities defined by Clinger–Cohen, the CIO is charged with ensuring that the information infrastructure will support full spectrum dominance. In order to do this the CIO must now address issues of interoperability and standardization.

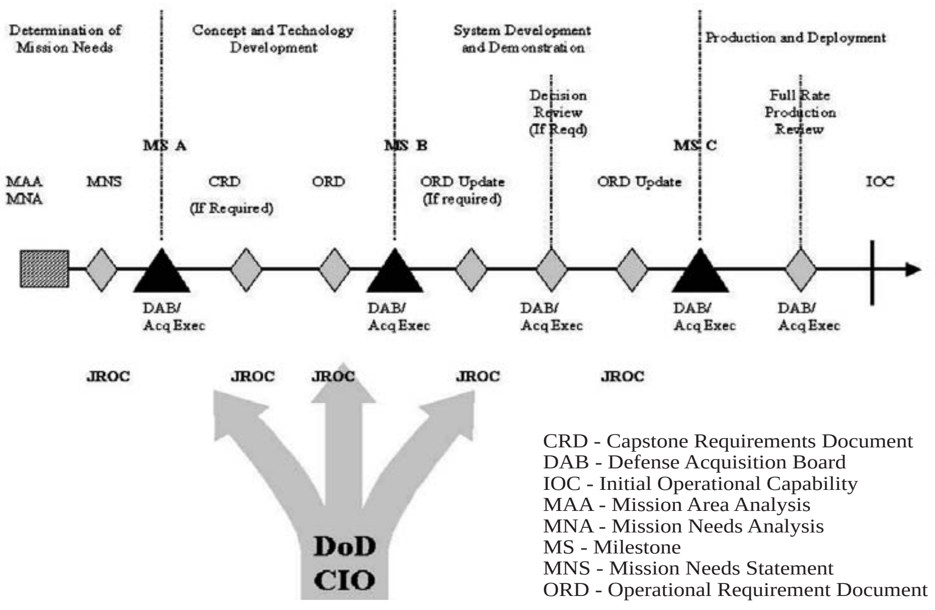
In October of 1998, legislation was enacted that added additional responsibilities for the DoD CIO and the CIOs of the military departments. Public Law 105-261, Strom Thurmond National Defense Authorization Act for fiscal year 1999 set forth, in addition to the responsibilities in the Clinger-Cohen Act, the following:

1. Review and provide recommendations to the secretary of defense on DoD budget requests for IT and NSS.
2. Ensure the interoperability of IT and NSS throughout the DoD.
3. Ensure that IT and NSS standards that will apply throughout the DoD are prescribed.
4. Provide for the elimination of duplicate IT and NSS within and between the military departments and defense agencies.

Ensuring interoperability of IT and NSS places the CIO in the forefront of building the foundation for JV 2020. The law also strengthens the CIO role in the requirements and acquisition process by providing a mechanism for budget review and recommendations.

To ensure a robust infrastructure, the CIO is a key player in requirements and acquisition process of the systems that will provide full spectrum dominance. Figure 1 depicts the DoD CIO's sphere of influence throughout the acquisition life cycle, which is ensconced in the charter of the Joint Requirements Oversight Council. "The Director of Architecture and Interoperability in the Office of the DoD CIO will serve the Joint Requirements Oversight Council (JROC) in an advisory role on Information

Figure 1: *The DoD CIO Role in the Acquisition Life Cycle*


Determination of Mission Needs
Concept and Technology Development
System Development and Demonstration
Production and Deployment
Decision Review (If Reqd)
Full Rate Production Review
MS A
MS B
MS C
MAA
MNA
MNS
CRD
(If Required)
ORD
ORD Update
(If required)
ORD Update
IOC
DAB/ Acq Exec
DAB/ Acq Exec
DAB/ Acq Exec
DAB/ Acq Exec
DAB/ Acq Exec
JROC
JROC
JROC
JROC
JROC
DoD CIO

CRD - Capstone Requirements Document
DAB - Defense Acquisition Board
IOC - Initial Operational Capability
MAA - Mission Area Analysis
MNA - Mission Needs Analysis
MS - Milestone
MNS - Mission Needs Statement
ORD - Operational Requirement Document

Defense Department CIOs

Department of Defense
Assistant Secretary of Defense (C3I)/Chief Information Officer
John P. Stenbit
http://www.c3i.osd.mil/

Joint Community
Director, Command, Control, Communications, and
Computer Systems (J-6)
LTG Joseph Kellogg
www.dtic.mil/jcs/ccrc/leadership.html

Department of the Army
Director of Information Systems for Command, Control,
Communications, and Computers (DISC4)
LTG Peter Cuviello
www.army.mil/disc4/index.html

Department of the Navy
Chief Information Officer
Mr. Dan Porter
204.222.128.9/doa-cio/cio-lib.html

Marine Corps
Director Command, Control. Communications,
and Computers (C4)
Brig Gen Robert F. Shea
issb-www.1.mqg.usmc.mil/cic/index.html

Department of the Air Force
Assistant Secretary for Acquisition
Dr. Lawrence Delaney
www.cio.hq.af.mil

Table 1: *Defense Department CIOs*

Technology, including National Security Systems. In addition, the DoD CIO will support JROC responsibilities for developing and validating the operational view of integrated operational concepts/architectures and related products as well as ensuring interoperability [8]."

As the chairman of the Joint Chiefs of Staff Instruction 3170.01b dated April 15, 2001 reiterates, the DoD CIO is responsible for ensuring the interoperability of IT and NSS throughout the DoD. The DoD CIO will ensure that IT and NSS standards that will apply throughout the department are prescribed, and provide for elimination of duplicate IT within and between the military departments and defense agencies. Through these recommendations, the CIO provides the advice to senior management personnel of the executive agency to ensure that IT is acquired in a manner that implements the policies and procedures of the Clinger-Cohen Act and Public Law 105-106.

The Global Information Grid (GIG) binds the architectural mandate of Clinger-Cohen with the interoperability and standards directives of Public Law 105-106. Through the GIG, the DoD CIO is implementing a sound and integrated architecture, which will provide globally interconnected information capabilities, associated processes, and personnel for collecting, storing, processing, disseminating, and managing information on demand to warfighters, policy-makers, and supporters [9]. The GIG supports all DoD, national security, and related intelligence community missions and functions (strategic, operational, tactical, and business) in war and in peace. The GIG provides interfaces to coalition, allied and non-DoD users, and systems [10]. The acquisition of GIG components will see a convergence of technology that will support the next leap in the revolution in military affairs and the revolution in business affairs.

Within the constructs of a defined architecture, the DoD CIO ensures the acquisition of interoperable systems that will set the foundation for full spectrum dominance. For the GIG to provide the right information at the right time to the right warfighter in the right format requires a high level of assurance. Thus it is no surprise that information assurance is one of the overarching policy considerations in the GIG architecture.

Assuring the Information Infrastructure

> "There's a war out there old friend, a world war, and it's not about who's got the most bullets. It's about who controls the information – about how we see and hear, how we work, what we think. It's all about the Information … [11]"

The United States possesses both the world's strongest military and its largest national economy. Those two aspects of our power are mutually reinforcing and dependent. They are also increasingly reliant upon certain critical infrastructures and upon cyber-based information systems [12]. As a result of advances in IT and the necessity of improved efficiency, the physical and logical separate systems of the infrastructures have become increasingly automated and interlinked. With an interoperable GIG as a foundation for achieving full spectrum dominance, successful implementation requires assurance. Presidential Decision Directive 63 tasks the CIOs with critical information infrastructure protection.

> Every department and agency of the federal government shall be responsible for protecting its own critical infrastructure, especially its cyber-based systems. Every department and agency CIO shall be responsible for information assurance. Every department and agency shall appoint a chief infrastructure assurance officer (CIAO) who shall be responsible for the protection of all of the other aspects of that department's critical infrastructure. The CIO may be double-hatted as the CIAO at the discretion of the individual department. These officials shall establish procedures for obtaining expedient and valid authorizations to allow vulnerability assessments to be performed on government computer and physical systems [12].

Within the DoD, the CIO is also designated as the CIAO. In most of the other federal agencies, a separate position or office of critical infrastructure protection has been created thus separating the protection of the information infrastructure from the physical infrastructure. However, in looking at the organizational structure under the DoD CIO, we find that reporting to the deputy assistant secretary of defense for Security and Information Operations are the Infrastructure and Information Assurance Directorate and the Directorate for Critical Infrastructure Protection. Thus, even within the DoD the policy organizations for the physical and information infrastructures are in distinctly separate office elements.

Thus with the issuance of Presidential Decision Directive 63, President Clinton placed the CIO at the forefront of information assurance and in some cases critical infrastructure protection within the federal government. For the critical networks within the information infrastructure and the information grid, security is a primary concern. Clinger-Cohen requires that the head of the executive agency shall ensure that the information security policies and practices of the agency are adequate. But, the act does not assign this responsibility for security directly to the CIO; that comes in subsequent legislation.

Linking Assurance and Security

The link between information assurance and information security is promulgated in the most recent piece of legislation expanding the responsibilities of CIOs. The Floyd D. Spence National Defense Authorization Act for fiscal year 2001 incorporates information security into federal information policy. In doing so, it spells out relationships between the CIO and head of the agency regarding the establishment of agency policy, procedures, and control techniques that will afford sufficient security protection commensurate with the risk. The law makes numerous references to the Clinger-Cohen Act of 1996 and establishes a link between tenets such as accountability, architecture, and security by delegating to the CIO the following authority:

1. Designate a senior agency information security official who shall report to the CIO or a comparable official.
2. Develop and maintain an agency-wide information security program.
3. Ensure that the agency effectively implements and maintains information security policies, procedures, and control techniques.
4. Train and oversee personnel with significant responsibilities for information security with respect to such responsibilities [13].

An important aspect of the legislation is that it requires the agency CIO, in coordination with senior agency officials, to periodically evaluate the effectiveness of the agency information security program, including the testing control techniques. The law further stipulates the each agency must develop an agency-wide information security program, and that the director of the OMB is tasked with approval and annual review.

The annual review process must be done with the program officials in consultation with the CIO. For the DoD and the Central Intelligence Agency (CIA), the approval and review authority rests with the secretary of defense and the director CIA. Finally, the law requires that each agency, in consultation with the CIO, shall include as part of the performance plan the resources and time required to implement the program based on a risk assessment of the agency.

This most recent piece of legislation forms the link between the security requirements of Clinger-Cohen and the assurance mandates of the Presidential Decision Directive 63. With this most recent legislation, we have seen that the responsibilities of federal CIOs have increased dramatically in a relatively short period of time. Today the CIO is facing such technology challenges as convergence and wireless, while providing assurance and security. To harness efficiencies across agencies, the question is begging: Does the federal government need an IT czar?

The Federal IT Czar

The speculation in Washington, D.C., is that President Bush will appoint a federal CIO. The Gartner Group has stated that "due to the transformational role of IT on government – Gartner believes that e-government transformation will eliminate at least 30 percent of the current government agencies. Gartner recommends that the new administration create a cabinet-level position within the executive office of the president to bring unity to the e-government movement. It is critical that the federal CIO be positioned as a key player in e-government and technology-related public policy. The president and the CIO should operate in tandem, much like successful chief executive officer (CEO)/CIO models in the private sector [14]."

Gartner Group is not the only lobby for the creation of this position. Sen. Robert Bennett (R-Utah) speaking at the U.S. Chamber of Commerce meeting "Cyber Security: The Real Y2K Challenge" stated, "The numerous legislative and agency efforts to address cyber security may need the guidance of a single 'chief information officer' to coordinate the government's cross agency and trans-industry security measures [15]."

From across the political aisle, Rep. Jim Turner (D-Texas) introduced legislation that would create the executive-level position and codify the executive order that created the interagency CIO Council [16]. There is bipartisan support as well, Sens. Fred Thompson (R-Tenn.) and Joe Lieberman (D-Conn.) chairman and ranking minority member on the Senate Governmental Affairs Committee, respectively, have expressed support for the concept [17]. Even with Congress, the GAO, and respected practitioners pushing, it is still questionable as to whether President Bush will establish an executive level IT position, a federal CIO.

Calls for a federal IT czar reinforce the criticality throughout government of the roles and responsibilities of CIOs. The continued explosion of IT in the revolution of government affairs places the CIO at the table with the CEO and the chief financial officer. The CIO ensures that technology provides seamless governmental operations and services. Facing issues in recruitment and retention, outsourcing, architecture, assurance, security, and resource management, clearly CIO does not mean, "career is over." To that individual who was sitting next to me at STC 1997, I hope you see the essential role of the CIOs in the age of e-government.

The views in this article are those of the author and do not reflect the official policy or position of the National Defense University, the DoD, or the U.S. government.◆

References

1. Baatz, E. B. "The Truth About Turnover." CIO Magazine. 1 Nov. 1996.

2. Section 5125. Agency Chief Information Officer. (b) General Responsibilities. Clinger-Cohen Act, Division E of the FY96 Defense Authorization Act, Public Law 104-106.
3. Section 5125. Agency Chief Information Officer. (c) Duties and Qualifications. Clinger-Cohen Act, Division E of the FY96 Defense Authorization Act, Public Law 104-106.
4. Section 5002. Definitions (3) Information Technology. Clinger-Cohen Act, Division E of the FY96 Defense Authorization Act, Public Law 104-106.
5. Section 5142. National Security System Defined (a) Definition. Clinger-Cohen Act, Division E of the FY96 Defense Authorization Act, Public Law 104-106.
6. Section 5141. Applicability to National Security Systems (b) Exceptions. Clinger-Cohen Act, Division E of the FY96 Defense Authorization Act, Public Law 104-106.
7. OSD Memorandum. Requirements for Compliance with Reform Legislation for Information Technology (IT) Acquisitions (Including National Security Systems), 1 May 1997.
8. Charter of the Joint Requirements Oversight Council, Chairman of the Joint Chiefs of Staff Instruction 5123.01A, 8 Mar. 2001.
9. OSD Memorandum. Global Information Grid, 22 Sept. 1999.
10. OSD Statement. Global Information Grid, 2 May 2001.
11. Sneakers. MCA Universal Pictures. 1992.
12. White Paper: The Clinton Administration's Policy on Critical Infrastructure Protection: Presidential Decision Directive 63, May 1998.
13. Section 3534. Federal Agency Responsibilities. Subchapter II–Information Security, SEC. 1061. Coordination of Federal Information Policy, Subtitle G–Government Information Security Reform, Title X, National Defense Authorization, Fiscal Year 2001, Public Law 106-398.
14. Caldwell, French, and Judith Carr, eds. Mr. President, Appoint a Federal CIO. Note Number: TG-12-8984, The Gartner Group, 18 Mar. 2001.
15. Krebs, Brian. "Federal CIO Needed for Web Security." Newsbytes, Daily News, Computer User.Com. 15 May 2001.
16. Trimble, Paula Shaki. "New push for federal CIO." Federal Computer Week 29 May 2000.
17. Wait, Patience. "Prospect of Federal CIO Still Lingers In the Wings." Washington Technology Vol. 15 No. 21. 5 May 2001.

About the Author

L. John Michel is a professor of Systems Management at the Information Resources Management College, the National Defense University. Currently, he manages the course "Assuring the Information Infrastructure." He formerly managed "Advanced Software Acquisition Management" and "Software Management for Executives." He has over 22 years experience in programmatic aspects of software intensive systems in the command and control, intelligence, and personnel communities. He has bachelor's and master's degrees in business administration from the University of Georgia.

Information Resources
Management College
National Defense University
Fort McNair, DC 20319
Phone: (202) 685-2062
Fax: (202) 685-3974
E-mail: michell@ndu.edu

Schnauzers, Modems, and Sound Techniques

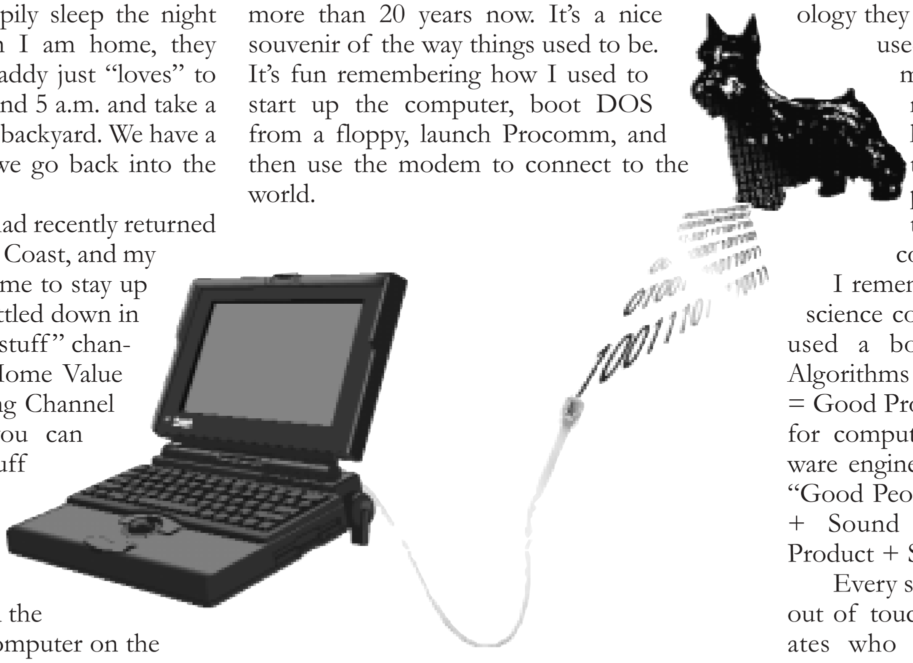
My wife and I are proud owners of two toy schnauzers. For you dog lovers out there, I can see the email already: “There is no such breed as a TOY schnauzer.” It’s true. We have two miniature schnauzers – both runts. Together, they weigh about 18 pounds. Trust me; they are TOYS.

Funny, when I am traveling and away from home, they happily sleep the night away. However, when I am home, they seem to know that Daddy just “loves” to get up with them around 5 a.m. and take a short walk around the backyard. We have a nice walk, and soon we go back into the house.

The other night I had recently returned from a trip to the East Coast, and my body thought it was time to stay up – so the dogs and I settled down in the den to watch the “stuff” channel. You know that Home Value Quality Stuff Shopping Channel (HVQSSC), where you can order all kinds of stuff that you didn’t know you needed. That night, the HVQSSC was trying to convince me that I needed the newest and greatest computer on the market. While I was listening, half-asleep, the announcer was discussing the computer’s features. Understand, now, that while white teeth, a dazzling smile and a smooth voice are necessities for hawking products on the HVQSSC – technical competence does not seem to be a prerequisite.

The announcer, in an excited voice, told me that the computer comes complete with a “Model 56K model manufactured by Baud.” Not sure I heard it right, I immediately hung up the phone (from ordering the all-in-one left-handed apple corer and spaghetti curler previously advertised) and listened intently. Sure enough, the announcer again said that the computer came with a Baud modem. That’s sort of like saying that a new car comes with a supply of miles per hour.

I used to be a modem expert. Long ago I owned several off-brand modems until I shelled out about $500 for a Hays 300 baud Smartmodem in 1982. Wow – 300 baud! Remember the Hays Smartmodem instruction set? To dial a number, you very carefully typed in “ATDT18005551212.” The “AT” meant attention, and the “DT” meant tone dialing, and the number followed. “ATDP” meant pulse dialing. “ATS1” returned a string that contained the modems’ current internal settings. I eventually graduated from the Hays 300 Smartmodem to a 1200, then a 2400. Still, the 300 was like a first childhood crush. I have kept that 300-baud modem for more than 20 years now. It’s a nice souvenir of the way things used to be. It’s fun remembering how I used to start up the computer, boot DOS from a floppy, launch Procomm, and then use the modem to connect to the world.



Now, of course, you don’t have to know how to use a modem. The knowledge I once mastered and was so proud of is now pretty well obsolete. Is my knowledge now useless? Far from it – the knowledge allows me to understand what a modem really does, and what the operating system does for me. I don’t have to access the modem directly anymore, but I understand how the modem actually works. I know how the modem integrates into the operating system, and if necessary, can use the control panel to debug and hopefully fix a malfunctioning modem. It’s old knowledge, but still useful even with modern computers and operating systems. The knowledge might be old, but it’s tried and true. It’s sound knowledge (pun intended!).

This issue of CrossTalk contains the winners of the Top 5 Quality Software Projects. I was privileged to be a reviewer for the projects, and was impressed with the quality of software that the Department of Defense puts out.

Did the projects succeed because they incorporated some new language, tool, or technique? No, not really. The projects succeeded because of three issues: good people, good management, and sound techniques. Some used modern languages and techniques, but all used pretty standard stuff. Some used languages that have been around 20+ years. Some didn’t even mention what development methodology they used – but you can bet all used some type of life-cycle methodology. All had a risk management plan. All collected metrics, discovered trouble spots, and took preventative actions to get the project out the door to correctly meet user needs.

I remember teaching a computer science course a few years ago that used a book that stated “Good Algorithms + Good Data Structures = Good Programs.” That might work for computer science, but for software engineering, the equation reads “Good People + Good Management + Sound Techniques = Quality Product + Satisfied Customer.”

Every so often, I start to feel a bit out of touch. I talk to recent graduates who understand Java Beans, Extreme Programming, SuperVisual J++++, and other tools that are not in my repertoire. Then I think “big deal.” Not one of the Top 5 projects said, “We couldn’t have completed the project on time and under budget if it had not been for the marvelous GUIs that came with the polymorphic inheritance package of SuperVisual Cobol ++.” Instead, requirements management, risk management, and project management made the project succeed – sound, tried-and-true techniques.

You want to go back to being a hacker? Then go take a course, and learn some new language that probably won’t be around in 10 years. What, you would rather get a project out the door on time, under budget? You prefer a satisfied customer? Then quit worrying about being a hacker and keep on concentrating on sound techniques. You aren’t obsolete – you’re just sticking with sound techniques that you know still work.

–David A. Cook,
Software Technology Support Center
david.cook@hill.af.mil

DYNAMIC WORKSHOPS SET IN THE PLAYGROUND OF THE WEST

The Air Force's Software Technology Support Center (STSC) is sponsoring eight workshops designed to help organizations build and buy software better. These informative workshops are **FREE** to U.S. government employees and will for the second year be held at the foot of the majestic Wasatch Mountains in state of the art facilities in the vicinity of Hill Air Force Base.

The series kicks off with **The Requirement for Good Requirements** on **January 15-17**. This workshop covers the fundamentals of requirements engineering, analysis, elicitation, documentation, and verification and validation. STSC consultants David Cook Ph.D. and Theron Leishman will utilize their many years of hands-on experience to show attendees how to get their requirements right the first time. The workshop also includes planned exercises to help participants solidify the concepts they learn.

The STSC workshop series continues throughout 2002 with these topics:

March 19-22	Process Improvement Using Theory of Constraints Thinking Process
April 24-26	Risk Management
May 21-23	Introduction to CMMI
June 18-20	Software Oriented Test and Evaluation
July 16-17	Software Best Practices: An Executive's Perspective
August 13-15	Object-Oriented Software Development: The Basics for Project Managers and Practitioners
September 17-19	Lifecycle Software Project Management from Systems Engineering to Post Development Support

Act Quickly.

Photos courtesy Utah Travel Council/Frank Jensen

For additional information visit our Web site at **www.stsc.hill.af.mil.**

SPACE IS LIMITED. To reserve your place at any of these workshops, contact Debra Ascuena at 801-775-5778 (DSN 775-5778) or debra.ascuena@hill.af.mil.

S T S C

MISSION CRITICAL SOFTWARE
CRSIP

Sponsored by the Computer Resources Support Improvement Program (CRSIP)

CROSSTALK / TISE
5851 F Ave.
Bldg. 849, Rm B04
Hill AFB, UT 84056-5713