

SPECIAL ISSUE: THE IMPORTANCE OF SOFTWARE

CROSSTALK

January 2004 *The Journal of Defense Software Engineering* Vol. 17 No. 1



Information From Senior Leadership

Information From Senior Leadership

4 Horizontal Fusion: Enabling Net-Centric Operations and Warfare
The Department of Defense's Horizontal Fusion Initiative will equip warfighters with the ability to access real-time information at the right time to make the right decisions.
by John P. Stenbit

7 Net-Centric Warfare Is Changing the Battlefield Environment
The Defense Information Systems Agency's net-centric technologies enable today's technologically superior warfighter to gain power from information, access, and speed.
by Lt. Gen. Harry D. Raduege Jr.

10 Military-Use Software: Challenges and Opportunities
The future vision for software within the Air Force must integrate software with other engineering disciplines, set a culture of professionalism, and employ an enterprise solution.
by John M. Gilligan

13 Charting the Course for the Department of the Navy's IM/IT Transformation
The Department of the Navy's chief information officer talks about the department's information management/information technology transformation agenda.
by David M. Wennergren

16 The Fire Support Software Engineering Division Achieves CMMI Level 5
These authors outline how this Army division became the first organization to attain CMMI Level 5 after 13 years of continuous pursuit and practice of a software and systems improvement program.
by Milton Smith and Phil Sperling



Best Practices

20 The AV-8B Team Learns Synergy of EVM and TSP Accelerates Software Process Improvement
This author explains the accelerating initiatives the AV-8B used to achieve Capability Maturity Model for Software Level 4 in 16 months versus the 50-month average reported by the Software Engineering Institute.
by Lisa Pracchia

Software Engineering Technology

23 Object-Oriented Layers in ELIST
This article discusses the object-oriented design strategies used to develop the latest version of the Enhanced Logistics Intra-Theater Support Tool used to determine the feasibility of an intra-theater course of action.
by Mary Ann Widing, Kathy Lee Simunich, Dariusz Blachowicz, Mary Braun, and Dr. Charles Van Groningen

Open Forum

27 Extreme Software Cost Estimating
This article describes a management-centric approach to predicting software development cost and schedule in a modern, or extreme, development environment.
by Dr. Randall W. Jensen

Departments

3 From the Publisher

6 Coming Events

12 Web Sites

15 Call for Articles

31 BACKTALK

CROSSTALK

PUBLISHER	Tracy Stauder
ASSOCIATE PUBLISHER	Elizabeth Starrett
MANAGING EDITOR	Pamela Palmer
ASSOCIATE EDITOR	Chelene Fortier-Lozancich
ARTICLE COORDINATOR	Nicole Kentta
CREATIVE SERVICES COORDINATOR	Janna Kay Jensen
PHONE	(801) 586-0095
FAX	(801) 777-8069
E-MAIL	crosstalk.staff@hill.af.mil
CROSSTALK ONLINE	www.stsc.hill.af.mil/ crosstalk

Subscriptions: Send correspondence concerning subscriptions and changes of address to the following address. You may e-mail or use the form on p. 19.

Ogden ALC/MASE
6022 Fir Ave.
Bldg. 1238
Hill AFB, UT 84056-5820

Article Submissions: We welcome articles of interest to the defense software community. Articles must be approved by the CROSS TALK editorial board prior to publication. Please follow the Author Guidelines, available at <www.stsc.hill.af.mil/crosstalk/xtlkguid.pdf>. CROSS TALK does not pay for submissions. Articles published in CROSS TALK 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 CROSS TALK.

Trademarks and Endorsements: This DoD journal is an authorized publication for members of the Department of Defense. Contents of CROSS TALK 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 CROSS TALK Editorial Department.

STSC Online Services: www.stsc.hill.af.mil
Call (801) 777-7026, e-mail: randyschreffels@hill.af.mil

Back Issues Available: The STSC sometimes has extra copies of back issues of CROSS TALK 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.



Changes Come to CROSSTALK and to Warfare Operations



It is with some modest pride and a little sorrow that I take this final opportunity to pen CROSSTALK's From the Publisher column to introduce this special issue, "Information From Senior Leadership." I have personally spent the past eight years as either the deputy or director of the Air Force Computer Resources Support Improvement Program. During the past 16 years, this program office has had the satisfaction of being the sponsor of both CROSSTALK and the Software Technology Conference. I would like to believe that these have become significant, respected venues for the communication and exchange of technology information relating to software and systems engineering within the Department of Defense (DoD).

However, as with all things in life, changes do occur. We are now embarking upon a new path for these activities involving a wider group of sponsors. Whereas CROSSTALK has formerly been funded exclusively by the Air Force, we are now in the process of extending that sponsorship to a broader spectrum of supporters. You will see some changes occur in CROSSTALK later in 2004 as we make that transition. In order to offset the publication and distribution costs of this journal, we will be including some advertisements from these new sponsors. I want to assure you that all attempts will be made to retain the best features of this respected, refereed journal as we transition to this new model of financial sponsorship.

In this issue, we are pleased to present the views from several of our chief information officers (CIO) within the DoD related to our new world of software-intensive systems. We begin with a view of horizontal fusion explained by John P. Stenbit, assistant secretary of defense for Networks and Information Integration and CIO of the DoD. In *Horizontal Fusion: Enabling Net-Centric Operations and Warfare*, Stenbit focuses on putting needed information immediately at users' fingertips to streamline business processes and win wars.

Next, *Net-Centric Warfare Is Changing the Battlefield Environment* by Lt. Gen. Harry D. Raduege Jr., director, Defense Information Systems Agency (DISA), is a comprehensive look at how the agency's contributions to net-centric warfare span all areas of the DoD. Raduege details how DISA's net-centric technologies and Internet sharing contribute to enabling today's technologically superior warfighter to gain power from information, access, and speed.

In *Military-Use Software: Challenges and Opportunities*, Air Force CIO John M. Gilligan says the future vision for software within the Air Force must focus on integrating software as a part of the system engineering discipline. He also stresses the importance of changing the culture of Air Force software professionals and managers to view software as simply one component of the larger information enterprise.

Similarly, the Department of the Navy CIO David M. Wennergren talks about what the Navy and Marine Corps teams are doing to answer the challenge to "deliver tomorrow, today." In *Charting the Course for the Department of the Navy's IM/IT Transformation*, Wennergren says the measure of an effective CIO is his or her ability to lead change.

Our final theme article demonstrates the U.S. Army's successful implementation of good software process, which sets a future goal to continually improve the software engineering capability of all U.S. Army software development agencies. In *The Fire Support Software Engineering Division Achieves CMMI Level 5*, Milton Smith and Phil Sperling discuss the path that led this Army organization to be the first in the DoD to achieve a CMMI® Level 5 rating.

As is usual, we also have a broad lineup of supporting articles that include the successful use of Earned Value Management and Team Software ProcessSM; a tutorial of object-oriented programming techniques in large, complex software systems; and extreme software cost estimating. This issue of CROSSTALK presents a broad diversity of timely information related to software development challenges and successes. It is hoped that the upcoming changes to the sponsorship of CROSSTALK will in no way diminish the ability to continue to bring you this critical information in a timely and reader-friendly fashion. I am pleased to have been associated with CROSSTALK for the past eight years and hope that my future assignments are as rewarding.

H. Bruce Allgood

Director, Computer Resources Support Improvement Program



Horizontal Fusion: Enabling Net-Centric Operations and Warfare

John P. Stenbit
Department of Defense



The terrorist attack on the Pentagon on Sept. 11, 2001 jeopardized America's command and control system at the very moment the system was most critically needed. Consequently, the U.S. Department of Defense has developed an initiative that enables the military to establish net-centricity, which is a global, Web-enabled environment that leverages existing and emerging technologies. Net-centricity makes it possible to move beyond traditional communities of interest such as command and control or intelligence, to full information exchange across the battlespace. The Horizontal Fusion Initiative is the user-oriented catalyst for net-centric transformation of the department. It will provide real-time situational awareness across the battle chain, allowing users to control and tailor needed information. Users will be able to broadly search or set preferences and subscribe to military operations and intelligence information that support their mission.

On Sept. 11, 2001, the unthinkable in the Department of Defense (DoD) happened. Five terrorists using an American Airlines plane as a weapon attacked the military headquarters of the world's sole superpower in an attempt to decapitate the most lethal military force on earth. While two similar attacks had just taken place in New York City, the attack on the Pentagon cut one of two major trunk lines into the Pentagon, jeopardizing America's command-and-control system at the very moment the system was most critically needed.

This incident called into question everything about how the military manages information. Should the military still have been based on hierarchical structures wired in series? Did every node in the system add value to the information,

or was much of the information needed raw but immediately by warriors in the field? Could warriors in battle effectively marshal sufficient data to perform the mission in an age of interconnected forces that needed rapid targeting decisions and zero friendly fire? Could warriors at the edge of the spear make sense of the data they got? The answers were transformational: horizontal fusion.

Horizontal fusion is a new initiative sponsored by the office of the DoD chief information officer (see Figure 1). It is a critical element in Secretary of Defense Donald Rumsfeld's vision of force transformation – to “think differently and develop the kinds of forces and capabilities that can adapt quickly to new challenges and to unexpected circumstances.”

An important factor in force transfor-

mation is *power-to-the-edge* – equipping warfighters across the entire battlespace with the ability to access needed information at the right time to make the right decisions. Power to the edge means making information available on a network that people can depend on and trust, and populating the network with new, dynamic sources of information to defeat the enemy while denying the enemy advantages and exploiting its weaknesses.

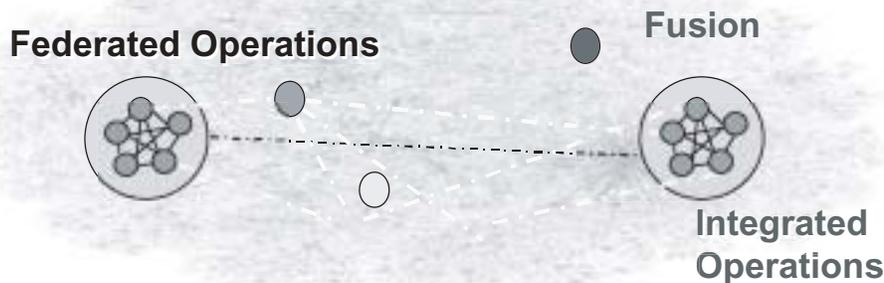
Achieving power to the edge means achieving net-centricity, which is a global, Web-enabled environment that leverages existing and emerging technologies. It assures user-focused information sharing, information fusion, sense making (of complex and ambiguous situations) and decision making across the battlespace. Net-centricity makes it possible to move beyond traditional communities of interest such as command and control or intelligence, to full information exchange across the battlespace.

A central benefit of net-centricity is the increased availability of information via the Task, Post in Parallel, Process in Parallel, and Use in Parallel (TPPU), or *smart pull* paradigm (see Figure 2). For TPPU to work, information must be posted immediately before it undergoes lengthy processing. The principle recognizes that users are smarter than their sources about what is needed operationally right now, unlike the legacy process, Task, Process, Exploit, and Disseminate.

Smart pull means that information is more accessible and gathered in smarter ways: cycle-time is in seconds; infrastructures are interoperable; real-time collaboration supports both standing and ad-hoc communities of interest; networks are robust; bandwidth is secure; operating

Figure 1: *What Is Horizontal Fusion?*

Horizontal fusion ensures that warfighters and analysts have timely and assured access to critical data and the leading-edge capabilities to make sense of that data.



Horizontal fusion is net-centric with the following:

- A focus on data and cross functional posting.
- Ad hoc access to and fusion of data that are created by operations that are both integrated and federated.
- A focus on making sense of that data.

mode is risk management vs. avoidance; and security supports and protects processes, not the other way around. The result is a warrior at the tip of the spear who can access critical information in real time – patrolling both their physical battlespace and the information cyberspace.

Net-centric transformation relies on these efforts:

- **The Global Information Grid (GIG) Bandwidth Expansion Program.** Provides a secure, robust, optical Internet protocol terrestrial network.
- **Joint Tactical Radio System.** A family of software-reprogrammable radios based on an open-communication architecture that will provide interoperable, tactical, wideband Internet protocol communications capabilities.
- **Wide-Band Satellite Communications.** Provides ubiquitous communications with optical quality bandwidth to mobile and tactical users.
- **Net-Centric Enterprise Services.** Provides the infrastructure services to support the broad range of applications and data used in a net-centric enterprise.
- **Information Assurance.** Supports all efforts to ensure that the net is robust, reliable, and trusted.
- **Horizontal Fusion.** Net-centric applications and content needed to provide analysts and warfighters with the ability to make sense of complex and ambiguous situations.

The Portfolio Concept

Horizontal fusion is the user-oriented catalyst for net-centric transformation of the DoD. It will provide real-time situational awareness across the battle chain, sense-making tools, and collaboration among multiple communities of interest and critical intelligence information sharing (see Figure 3).

The 2003 Horizontal Fusion Initiative integrates advanced technologies to make the *quantum leap* to net-centric operations, emphasizing support to warfighters located at the edge of the GIG. The objectives for the 2003 Horizontal Fusion Initiative are computing at the edge, publishing information to the GIG, sharing intelligence and surveillance and reconnaissance data in the DoD and the intelligence community, improving operational-intelligence data interoperability, and exploiting many diverse data sources and providing the tools to make sense of

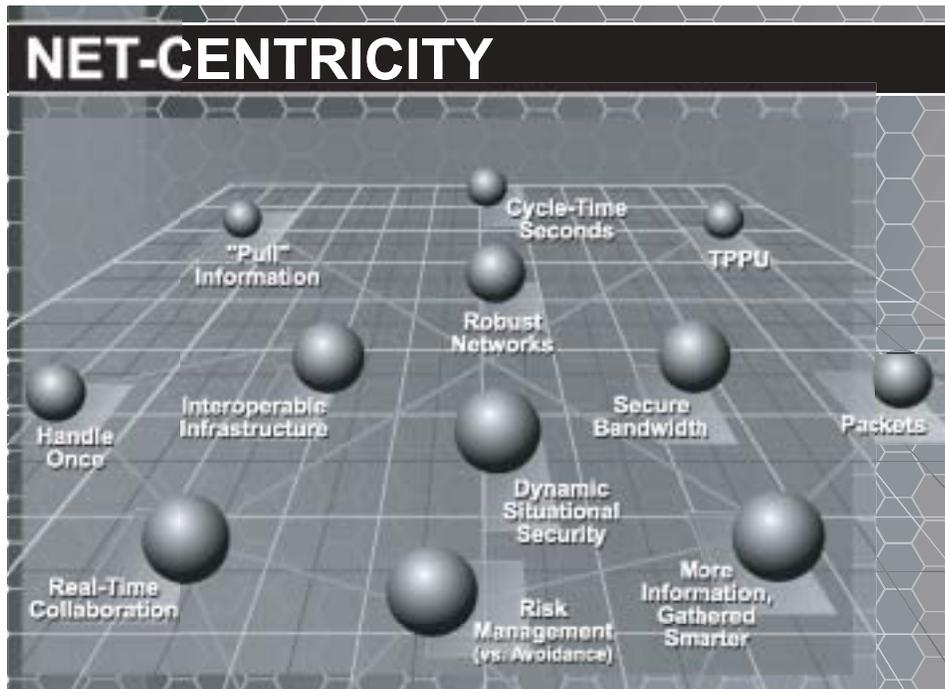


Figure 2: Net-Centricity Paradigm

the data.

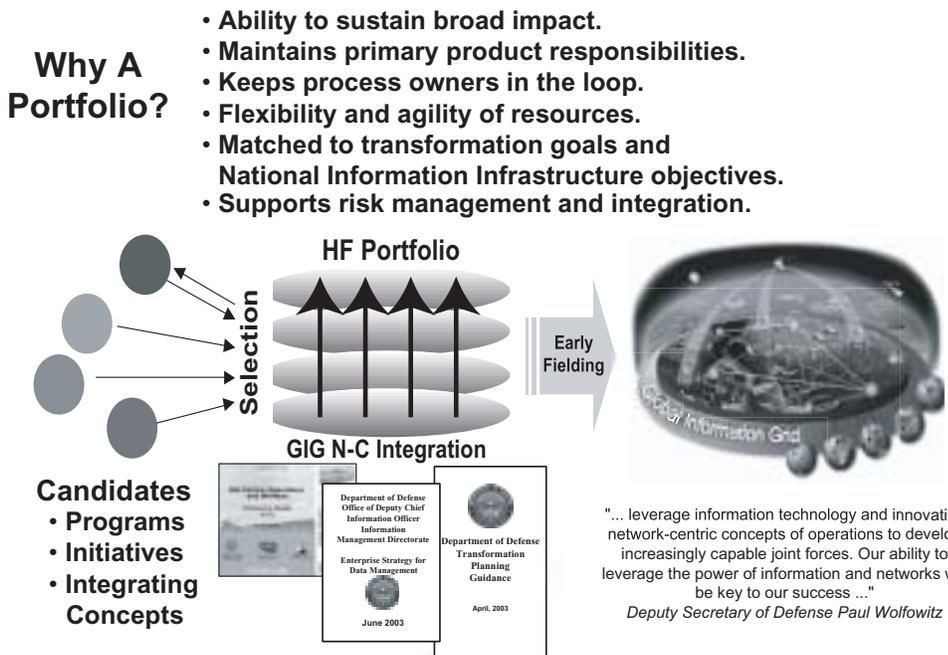
Horizontal fusion is not a single program, but a portfolio of net-centric initiatives. Using a common architecture and integration process, these initiatives are woven into an information tapestry called the Collateral Space, which is accessed via a portal. The portal's main characteristic is that users can control and tailor the pull and portrayal of information. Users are able to broadly search or set preferences and subscribe to military operations and intelligence information that supports their mission.

The 2003 Horizontal Fusion

Quantum Leap-1 (QL-1) effects-based assessment and demonstration involves warriors at the edge of the network who can tap various communities of interest and achieve the speed of command and performance improvement needed to neutralize a time-critical target. The scenario for QL-1 was chosen to assess the value of the Collateral Space as the warriors' ready source of situational awareness in a net-centric environment. All capabilities successfully demonstrated remain in place and available for operational use.

Horizontal fusion does not end with

Figure 3: Horizontal Fusion's Portfolio Concept



COMING EVENTS

February 1-4

*3rd International Conference on
COTS-Based Software Systems*
Redondo Beach, CA
www.iccbss.org/2004

February 2-7

*IT Service Management Forum
Conference and Expo*
Washington, D.C.
www.jupiterevents.com

February 3-5

*WEST 2004
Western Conference and Exposition*
San Diego, CA
www.west2004.org

March 1-3

*17th Conference on Software Engineering
Education and Training (CSEET 2004)*
Norfolk, VA
www.cs.virginia.edu/cseet04

March 8-11

*Software Engineering Process
Group Conference 2004*



Orlando, FL
www.sei.cmu.edu/sepg

March 29-April 1

*Defense Technical Information Center
Annual Meeting and Training Conference*



Alexandria, VA
www.dtic.mil/dtic/annualconf

March 30-31

*3rd Annual Southeastern Software
Engineering Conference*
Huntsville, AL
www.ndia-tvc.org/SESEC

April 19-22

2004 Software Technology Conference



Salt Lake City, UT
www.stc-online.org

QL-1; activities are programmed through 2008. In 2004, we will concentrate on expanding to other communities of interest with the Collateral Space and piloting additional enterprise services. Cross-domain information sharing and secure wireless communications are major investment areas. We will continue to add edge users and data sources to the Collateral Space. Working with the intelligence community, we will demonstrate cross-domain information sharing and collaboration in QL-2.

As the Horizontal Fusion Initiative progresses, it will be collaborative and contribute to other transformational efforts such as the Office of Force Transformation (focused on Force Transformation), and Joint Forces Command (focused on inter-service interoperability), as well as current and emerging efforts to transform warfighting and intelligence paradigms into 21st century realities.

The horizontal fusion portfolio will continue to provide value to the warfighters in several ways:

- Incorporating and tagging data from all sources and allowing it to be seen and used in innovative ways.
- Providing sense-making tools to analyze and understand this diverse and immense data set.

- Assuring that data pulled are qualitative, not quantitative.
- Achieving rapid insertion of tools and capabilities that will implement net-centricity across the department.
- Leveraging legacy investments while influencing future investments and introducing new technologies.

With these activities, the overarching goal of horizontal fusion is to be the catalyst for net-centric transformation of the department. It will support DoD and the intelligence community in accelerating efforts to achieve superiority in the transformed battlespace.

It can be summed up this way: By placing the information that is needed immediately at the fingertips of our people who need it, horizontal fusion will revolutionize how America wins wars, and greatly streamline our business processes. No one node of our system of systems will be the chink in our armor allowing an enemy to inflict a mortal wound. No enemy will get inside our observe-orient-decide-and-act loop, because there will be no challenger who can harness facts faster or act more decisively. Horizontal fusion gives us the power to prevail in an age of net-centricity. It will transform the department just as it will transform the world. ♦

About the Author



John P. Stenbit is assistant secretary of defense for Networks and Information Integration and chief information officer of the Department of

Defense. He has more than 30 years of public- and private-sector service in the telecommunications and command and control fields. This includes two years as principal deputy director of Telecommunications and Command and Control Systems for the Department of Defense, and two years as staff specialist for Worldwide Command and Control Systems in the Office of the Secretary of Defense. Previously, Stenbit was an executive vice president of TRW and was responsible for the planning and analysis of advanced satellite surveillance systems. Prior to that, he was at the Aerospace Corporation and was involved with command and control systems for missiles and satellites, and satellite data compression and pattern recognition.

He has chaired the Science and Technology Advisory Panel to the director of Central Intelligence, and served as member of the Science Advisory Group to the directors of Naval Intelligence and the Defense Communications Agency. He also chaired the Research, Engineering and Development Advisory Committee for the administrator of the Federal Aviation Administration. He has served on the Defense Science Advisory Board, the Navy Studies Board, and the National Research Council Manufacturing Board. In 1999, Stenbit was inducted into the National Academy of Engineering. He has bachelor's and master's degrees in electrical engineering from the California Institute of Technology.

**ASD Networks and
Information Integration
6000 Defense Pentagon
Room 3E172
Washington, D.C. 20301-6000**

Net-Centric Warfare Is Changing the Battlefield Environment

Lt. Gen. Harry D. Raduege Jr.
Defense Information Systems Agency



U.S. military forces today are creating and executing plans using capabilities that were not available as recently as Operation Desert Storm. This is due to net-centric warfare and the information transfer and sharing that is made available by the Internet. Today, the nation's armed forces, armed with superior technology, gain power from information, access, and speed. This article presents these new capabilities and outlines how the Defense Information Systems Agency's contributions to net-centric warfare span across all areas of the Department of Defense.

Net-centric warfare is not just about technology; it is an emerging theory of war and the next art and science of warfare to be exploited. Net-centric warfare involves a cultural change in relationships that includes networking over the Internet among large groups of people. America's armed forces are now creating and executing plans using capabilities that were not available 12 years ago during Operation Desert Storm in Iraq when the military advantage still came from numbers of platforms and people in the battlespace. Today, our nation's military forces, armed with superior technology, gain power from information, access, and speed.

Air Force Gen. Dick Myers, chairman of the Joint Chiefs of Staff, identified "the application of force, using forces in an integrated way, and having the eyes, ears, and command and control to carry it off" as the most important factors in Operation Iraqi Freedom (OIF). This is also the core of net-centric operations.

Net-centric warfare combines a powerful military force with information superiority, giving American service men and women greater awareness of our own forces, the enemy, and the battlefield environment. America now has a smaller, more lethal deployed military force. Net-centric operations permit forces to focus on specific targets, protecting the lives of American and coalition forces, as well as countless non-combatants.

"With less than half of the ground forces and two-thirds of the air assets used 12 years ago in Desert Storm, we have achieved a far more difficult objective ... In Desert Storm, it usually took up to two days for target planners to get a photo of a target, confirm its coordinates, plan the mission, and deliver it to the bomber crew. Now we have near real-time imaging of targets with photos and coordinates transmitted by e-mail to aircraft already in flight. In Desert Storm, battalion, brigade, and division commanders

had to rely on maps, grease pencils, and radio reports to track the movements of our forces. Today, our commanders have a real-time display of our armed forces on their computer screens," said Vice President Richard Cheney.

Much of the United States' success during OIF is due to tremendous advancements in the world of informa-

"Both the secretary [of defense] and the chairman used GCCS reports to brief the president on operations and force locations in and around Iraq. This marked the first time a common operational picture was available at all levels ..."

tion sharing and situational awareness, for both U.S. and coalition forces. This enables essential command, control, communications, and intelligence components. Such technology advancements, many of which the Defense Information Systems Agency (DISA) developed and/or supported, include the following:

- The DISA-operated Defense Information System Network (DISN) carries the vast majority of the Department of Defense (DoD) telecommunications. As such, the DISN provides global classified and unclassified voice, data, video, and transmission services through predominantly commercial assets supple-

mented with military value-added features. Those military features provide greater global reach, security and encryption options, interoperability, and high levels of reliability. These features ensure that U.S. forces are not denied access to critical information, geography, or battle space. In Operation Enduring Freedom (OEF) in Afghanistan in 2001 and OIF, there was a literal explosion in the demand for bandwidth by deployed forces. More than 50 times more bandwidth was used per person in OIF than in Desert Storm. Greatly expanded bandwidth, voice, and data capacity combined with an impressive set of early net-centric capabilities allowed Army Gen. Tommy Franks and his battle staff to collaborate, plan, and execute their mission with a smaller footprint forward with virtual support from rear assets. When full-up hostilities began in the U.S. Central Command (USCENTCOM) area of operations, deployed forces had what they needed to support the myriad of systems military commanders used to control forces on land, sea, and air. Through advanced planning, U.S. forces also had the requisite bandwidth for voice, data, and imagery.

- The Global Command and Control System (GCCS) provided a Common Operational Picture (COP) across military service lines for near-instantaneous command. Since the global war on terrorism started, and has continued through OIF, DISA has successfully upgraded the GCCS software 27 times. Those upgrades were accomplished while the system remained fully operational, serving the needs of all nine combatant commanders. In response to a request from the commander, USCENTCOM, DISA also accelerated the delivery of a key intelligence capability several months early. In OIF, the improved intelligence and

Operation Iraqi Freedom Successes

- First installation of fiber optics into Southwest Asia – 138x increase to 555 Mbps.
- Ubiquitous commercial satellite communications (SATCOM) to supplement military SATCOM – 10x increase to 3200 Mbps.
- Data network expansion – 6x increase to SIPRNET and NIPRNET to 130 Mbps.
- First all-service, Special Operations Forces, red, blue, and intelligence-fused picture.
- Extensive coordinated use of unmanned aerial vehicles to include supporting Command, Control, Communications, Computers and Intelligence networks.
- First real-time in-transit visibility plus logistics queries in minutes instead of hours.
- First use of interoperable desktop collaboration tools for C2.
- First widespread use of VTC as a C2 system in wartime – 22x increase in conferences since 9-11.
- Extensive coalition information sharing.
- First use of record copy traffic with attachments.
- 39x increase in voice (Defense Switched Network) and 5x increase in Defense Red Switch Network.

imagery capability and availability of Army ground force information on the network provided truly joint situational awareness for the first time that included all military services, red, blue, Special Operations Forces, and intelligence information for the warfighter. These COP and Common-Intelligence Picture capabilities provided a crucial enhancement to the sensor-to-decision-maker-to-shooter requirements. GCCS Version 4.0 is on track for delivery in 2004. About 25 percent of GCCS is Web-enabled today. That will increase to about 50 percent with the GCCS 4.0 upgrade. DISA is also partnering with U.S. Joint Forces Command to transform the joint deployment process.

- DISA's Joint Staff Support Center installed GCCS terminals for both the secretary of defense and the chairman of the Joint Chiefs of Staff. Both the secretary and the chairman used GCCS reports to brief the president on oper-

ations and force locations in and around Iraq. This marked the first time a common operational picture was available at all levels from the president down to the task force commanders.

- The Global Combat Support System (GCSS) is another success story. A DoD public key infrastructure-enabled service and portal environment, GCSS provided feeds from a variety of logistics systems and was integrated with GCCS. USCENTCOM directed that all materiel resources flowing to the theater be monitored through the In-Transit Visibility (ITV) system. In support of OIF, DISA installed a network guard that moved unclassified information to the Secret Internet Protocol Router Network (SIPRNET). Queries that had previously taken hours were available in minutes – including ITV information – on the command and control network. GCSS queries increased more than 17 times from about 175 queries per month in September 2001 to more than 3,100 queries per month during OIF.
- Extensive collaboration was another huge new global war on terrorism initiative. DISA supported USCENTCOM's major command and control business process reengineering effort with a variety of collaboration capabilities. The USCENTCOM commander and his staff used DISA-provided secure video teleconferencing (VTC), as well as desktop collaboration with the Defense Collaboration Tool Suite (DCTS) at unprecedented levels and on a 7 x 24 basis. VTC, a huge consumer of bandwidth, proved to be a significant driver behind theater bandwidth upgrades in support of OEF and OIF. Deployed forces used the whiteboard, chat, and shared file capability in DCTS extensively. USCENTCOM discouraged desktop VTC, however, to reduce the impact on limited SIPRNET bandwidth.
- The Enhanced Mobile Satellite Service (EMSS) experienced exponential growth during the global war on terrorism and OIF. EMSS provides 7x24 global satellite phone and data coverage. Since Sept. 11, 2001, the number of users increased by 344 percent and usage increased by 4,800 percent to more than 2.57 million call minutes per month. This system allowed Special Operations Forces to call in air strikes from horseback in Afghanistan by permitting instantaneous communications in areas without any infrastructure whatsoever.

Net-centric warfare's effectiveness has greatly improved in 12 years. Desert Storm forces, involving more than 500,000 troops, were supported with 100 megabits per second (Mbps) of bandwidth. Today, OIF forces, with about 350,000 warfighters, had more than 3,000 Mbps of satellite bandwidth, which is 30 times more bandwidth for a force 45 percent smaller. U.S. troops essentially used the same weapon platforms used in Operation Desert Storm with significantly increased effectiveness.

DISA's contributions to net-centric warfare span across all areas of the DoD. When the president needs to talk with anyone in the world, at any security level, the White House Communications Agency is with him at all times every day of the year. When someone searches the Web for information on a particular piece of military equipment, chances are they are looking at a page from the Defense Technical Information Center. If a non-commissioned officer deconflicts frequency spectrum issues in Iraq or Afghanistan among the military services and their equipment, that officer probably works at DISA's Joint Spectrum Center. When a Navy F-14 flies up to an Air Force KC-10 and talks to the boom operator, DISA's Joint Interoperability Test Command already ironed out any wrinkles associated with multi-service communications connections. Obviously, these DISA organizations have a unique and essential role in America's defense.

Although DISA's focus remains the warfighter, it has received taskings to facilitate command, control, and coordination between DoD and non-DoD elements. The Defense Red Switch Network (DSRN), a secure voice capability, was established more than 10 years ago to support the White House, Joint Staff, combatant commanders, and other critical command and control (C2) users. It is now being expanded to include 18 additional federal government agencies in support of numerous homeland defense security initiatives. During the space shuttle Columbia recovery operations, U.S. Northern Command required immediate VTC to coordinate actions between 23 sites on a Saturday morning. Team DISA was able to respond to the situation and provided needed service during the emergency operation.

At DISA, we take our warfighter support job very seriously. We recognize we cannot rest on past successes so we are also preparing for the future – integrated information *on demand*. Products and services provided by DISA in support of OIF and OEF demonstrate that we clearly understand that we must be able to surge

the backbone and deliver joint and interoperable services globally and on demand. We are focused on that path of support. We recognize the significant challenges we face in information networking and providing power to the edge. We have developed a strategy to continue transforming DISA to meet the transformational demands of revolutionizing warfare. Air Force Gen. Ralph E. Eberhart, commander of NORTHCOM, has noted those challenges. He recently said, "We are usually pretty good at sharing information vertically. But we need technology that can share information horizontally."

The stove-piped systems of today with limited interoperability must be replaced with a secure, robust, intelligent, and interconnected nodal network of tomorrow. *Power*, in the form of *quality information* for individual warfighters on the front lines – wherever they are – must be made available to provide a synchronized, real-time vision of the battlespace with lightweight Web-based tools to facilitate planning and execution.

A representative sample of some of our efforts include support of the Transformational Communications Study (TCS), the Standardized Tactical Entry Point (STEP) migration to DoD Teleports, Global Information Grid Bandwidth Expansion (GIG-BE), GIG Enterprise Services (GIG-ES), and Joint C2.

A robust, integrated telecommunications infrastructure is a must for future warfare. The TCS seeks to architect the future communications satellite constellation by removing bandwidth as a consideration and moving to a seamless, end-to-end network information sharing environment supported by high-speed, high capacity, and interoperable communications. DISA has had and will continue to have a major role in the TCS effort. In addition to providing requirements analysis and architectural engineering support, DISA also performs the challenging task of transition analysis.

STEPs were used extensively during OIF. Tomorrow's DoD Teleports will far exceed today's STEP capabilities.

The DoD Teleport program, an initiative to increase DISN capability, allows deployed forces to connect through teleports to a multitude of commercial satellite frequencies. DoD teleports will be telecommunications collection, access, and distribution points that provide deployed warfighters with multi-band, multimedia, and worldwide reach-back that far exceed current capabilities. To meet today's combatant commanders' immediate needs, DISA has accelerated

the fielding of DoD teleports with IOC being reached last summer.

The GIG-BE will create a trusted ubiquitous *bandwidth-available* environment to improve national security intelligence, surveillance, reconnaissance, and command and control information sharing. The GIG-BE initiative brings high-speed bandwidth to numerous key locations globally, and will connect approximately 102 key intelligence, command, and operational locations with a state-of-the-art optical mesh network. DISA is currently working with the military services, combatant commands, and agencies to ensure that the resources provided by GIG-BE are optimized.

GIG-ES is an exciting new arena for DISA. It is envisioned as the virtual place where information can be integrated to make net-centric warfare possible. GIG-

"The stove-piped systems of today with limited interoperability must be replaced with a secure, robust, intelligent, and interconnected nodal network of tomorrow."

ES will provide us with a new way of thinking about and providing transformational C2 services to joint forces. GIG-ES will replace *legacy* platform-centric systems with net-centric concepts using a web-enabled, data centric *power-to-the-edge* construct. It builds upon the Defense Information Infrastructure Common Operating Environment (DII COE) to provide a tailorable services approach built upon a robust communications capability.

Just as the DII COE is morphing to GIG-ES, we expect a similar transformation for GCCS to Joint Command and Control (JC2) transformation. JC2 will employ a secure, collaborative, Web-enabled and tailorable command-and-control architecture and *capability packages* that provide decision superiority as well as vertical and horizontal interoperability. We expect JC2 to take advantage of GIG-ES services as they mature. Users will access fused information sources through common IP-based network services, common data representations, and common catalogs/directories using intelligent, thin, and

ubiquitous (e.g., wireless, personal decision assistant-type) clients. The JC2 Operational Requirements Document made its way through the Joint Requirements Oversight Council last year. We anticipate heavy DISA involvement in the JC2 Analysis of Alternatives.

The DISA team is very proud of its warfighter support over the past two years. But that will never be good enough. There are many challenges ahead: new technology, new business processes, and expanded partnerships. With a foot firmly in the present to sustain and improve operational capability, we have put our transformation foot forward as we move to net-centric warfare developments of the future. ♦

About the Author



Lt. Gen. Harry D. Raduege Jr. is director, Defense Information Systems Agency, Arlington, Va. As director, he leads a worldwide organization of more than 8,200 military and civilian personnel. This organization engineers, develops, acquires, and provides integrated command and control and information networks to serve the needs of the president, secretary of defense, joint chiefs of staff, the combatant commanders, and other Department of Defense components under all conditions ranging from peace through war. Raduege is also responsible for operating the most complex and far-reaching military information networks in the world. He entered the Air Force in 1970 through the Air Force Reserve Officer Training Corps program at Capital University, Columbus, OH. Prior to assuming his current position, he was the director of command control systems, Headquarters North American Aerospace Defense Command and United States Space Command, and director of communications and information, Headquarters Air Force Space Command. He also served as the chief information officer for all three commands.

Point of Contact:
Betsy Flood
DISA Public Affairs
Phone: (703) 607-6048
DSN: 327-6048
E-mail: floodb@ncr.disa.mil

Military-Use Software: Challenges and Opportunities

John M. Gilligan

Department of the Air Force



As the Air Force chief information officer, I focus on improving the management and application of information technology (IT) to improve decision-making in support of all Air Force missions. Increasingly, the effectiveness of our war-fighting capabilities is dependent on software, as most functionality in today's mission and support platforms is enabled by software. In the case of many IT systems, the software is the system. A collective challenge for government and industry is to improve our ability to rapidly develop and acquire robust software that meets today's needs and can evolve to support tomorrow's mission requirements.

Software has become the key element in enabling the military to field warfighting and combat support capabilities, and the importance of software will only increase in the future. The inherent characteristics of software provide three benefits for addressing military requirements: speed, linkage, and adaptability. Well designed and implemented software can be changed in minutes or hours versus the weeks and months required for hardware modification, thus permitting flexibility to address rapidly evolving mission requirements.

In addition, software governs most of the interfaces of today's systems. The flexibility of software enables us to quickly integrate separately developed systems permitting what our Air Force Chief of Staff Gen. John P. Jumper calls horizontal integration of our systems. We will achieve the vision of horizontal integration primarily through effective software application.

Finally, through software adaptability and linkage characteristics, software enables systems to adapt to new environments, new threats, and new concepts of operation. This adaptability is a key enabler to reaping the benefits of rapid technological change and providing the transformational battlefield envisioned by Joint Vision 2020¹. With software's inherent flexibility come significant challenges that we must actively address if we are to realize the many potential benefits of software.

Unfortunately, many people believe that anyone can write reliable software – it only takes a little creativity. Creativity is important to problem solving when using software; however, building a software system for use in the military environment that is reliable and can be maintained at a reasonable cost requires the application of rigorous engineering discipline. Unfortunately, the software industry has not consistently exhibited this discipline in the design, construction, and testing of systems.

The consequence is that we have many commercial products that are riddled with logic flaws, which decrease the reliability of the target system, and the flaws are increasingly becoming the targets of system attacks using viruses and worms. Moreover, in some instances, we have seen that the inherent flexibility of software

"... the inconsistent application of engineering discipline to software development has resulted in a very mixed track record for software-intensive projects and disappointment and distrust on the part of military customers."

coupled with weak understanding of the required engineering disciplines has led to unrealistic expectations of what can be accomplished for software components of weapon systems. Finally, the fact that software can be easily modified has sometimes led to a desire to change requirements in mid-project without assessing architecture impacts and bypassing steps in the software engineering process.

In short, the inconsistent application of engineering discipline to software development has resulted in a very mixed track record for software-intensive projects and disappointment and distrust on the part of military customers. As one Air Force four-star general recently put it, "I am a strong supporter of the benefits of

information technology for improving our war-fighting capabilities, but I worry if the network will be available when we need it."

To overcome these problems of the past, I suggest that our industry partners and we in the military take a fresh look at our software development paradigms and processes. Specifically, I recommend that we focus on three areas that I submit can help improve our software development and support to deliver tomorrow's critical capabilities, bridging the current gap between expectations and delivery. These three areas discussed below are as follows: *integration* of software with other engineering disciplines, establishing a *culture* of software professionalism, and employing an *enterprise* solution focus.

Integration

In the past, software was often viewed as an upstart technology – a black art understood by many and mastered by few. As a result, we tended to create a separate infrastructure for dealing with software issues. We had separate software policies, processes, and organizations. In effect, we created a software stovepipe. In reality, individual software solutions must be integral to and tightly integrated with all components of a system, or in most cases with the *system of systems*. We need to integrate software into our overall systems engineering processes. Software must be an integrated part of our acquisition and engineering policy, processes, training, and metrics.

One promising solution is the Capability Maturity Model[®] Integration (CMMI[®]) as a disciplined approach to system development and process improvement. As the model is extended to acquisition activities, we will build on the groundwork and lessons learned from software development to address our management and technical responsibilities.

This integration of software into our overall system engineering processes requires actions by the acquisition, com-

[®] Capability Maturity Model, CMM, and CMMI are registered in the U.S. Patent and Trademark Office.

munications and information, and software communities. We must first ensure that software-knowledgeable personnel understand and apply systems engineering practices. Software must be recognized and managed as an engineering discipline. Likewise, software personnel, both government and contractor, must make the effort to integrate their knowledge and practices into the current acquisition and engineering practices and policies. Software technical jargon, software *geek-speak* if you will, and vague explanations of software issues can hinder progress. While software professionals have a responsibility to educate others on software issues, they also have a responsibility to make terminology, practices, and tools understandable to non-software practitioners: managers, system engineers, and customers.

To help effect this change within the Air Force, we are just beginning to institute the Air Force Software-intensive Systems Improvement Program (AFSSIP). The AFSSIP is predicated on an understanding of software as both a capability enabler and a potential risk area to be identified, and addressed as integral to our overall system engineering processes and disciplines. The success of the AFSSIP requires a new culture with a strong emphasis on training and education.

Culture

Integrating software capability into our existing practices requires a culture change that explicitly recognizes the importance of software and strategic employment of software-knowledgeable people in today's Air Force. No longer can we just say, "Don't worry, we'll fix it in software," and proceed to sign up to unreasonable estimates of software effort, or be surprised when major system problems manifest themselves during software integration and testing. We must move from reactive to proactive in managing software as a capability enabler and a potential system engineering risk area to be identified and managed. As one example, we've established the Air Force Software Steering Group to proactively address software issues at the Headquarters Air Force.

To ensure proper application of software knowledge, we must ensure that software and systems engineering education and training are robust and available to a broad range of our personnel. This renewed emphasis on education and training applies not just to personnel overseeing development of software, but also program managers, system engineers, and even system operators. With a software-knowledgeable work force, we can ask the

right questions about software, better understand software impacts, and make decisions that are consistent with the state of the practice. This will result in the fielding of better software intensive systems. To this end, the Center for Systems Engineering at Wright-Patterson Air Force Base has been established to promote education, training, research, and consultation throughout the Department of Defense in the best practices of systems engineering, including software². In the future, we will focus on extending this systems and software engineering emphasis to our program management curricula and senior leader development programs.

We also need government software professionals to be involved in the highest leverage activities in the acquisition of software capability. We must instill in our software professionals knowledge of systems engineering disciplines, including robust architectural design, as well as the expertise necessary for insight into the engineering activities of our contractor partners. We must recognize that software coding, while enjoyable and rewarding, is just one small piece of the larger systems and software engineering discipline. We need software professionals that understand the *entire* process well enough to ensure that the software being developed for military use will be effective and reliable within the environment and enterprise that it must operate.

Enterprise

In today's network-centric battlefield environment, it is clear that no single system or platform provides the full set of capabilities required by a warfighter. As a result, integration among systems becomes a key focus, and seamless connections between systems become a primary requirement. A part of our objective is to leverage machine-to-machine communication – that is, letting computers automatically retrieve, exchanging, and analyze information against established patterns or criteria, thus relieving our operators of this burden.

At the engineering level, these connections require software that is designed and built to facilitate integration into a global enterprise of interconnected systems and information. Our military-use software must be interoperable and highly available using current and emerging technologies such as XML and Web services. Our ability to engineer systems to permit immediate integration with future systems as the environment changes becomes a lynchpin to supporting the battlefield of tomorrow.

In Fred Brooks' seminal essays on soft-

ware engineering, "The Mythical Man-Month" [1], he described a tenfold difference between making software that works (like what some of us may have written in an entry level programming course) and software that is integrated with all elements of the operating environment and hardened for rigorous use. His point was that robust software takes much more time and energy. Likewise, integration of software solutions into today's complex system-of-systems enterprise environment requires more effort and a focus on the integrated enterprise as the target environment.

Clearly, if the focus is not placed on the enterprise from the beginning of the software design, as with Brooks' example, it may require 10 times the effort later in the life cycle to enable the software to effectively perform in our horizontally integrated, network-centric environment. It is incumbent upon the software professional to understand the bigger target picture for any individual software solution. Clearly, this will become a critical success factor for our software intensive systems.

The use of architecture is fundamental to help in achieving an integrated enterprise vision. Architectures enable us to understand and visualize mission and system relationships and to manage the complexity of developing integrated systems. To realize the benefits of architectural engineering and avoid ineffective, duplicate, and costly systems, we must ensure our software is engineered to satisfy operational architecture requirements and within the context of the appropriate system and technical architectures.

One example of effective enterprise architecting is the Air Force's Global Combat Support System (GCSS-AF) Integrated Framework³. The GCSS-AF Integrated Framework architecture provides core enterprise services to all applications, thus reducing the cost of developing and integrating applications while promoting standards for security and interoperability. To date over 60 key logistics applications are accessible through the framework, drawing nearly half a million hits per day.

Conclusion

Software is a critical enabler to achieving today's warfighting and combat support capabilities. As we transform towards tomorrow's net-centric future, proper engineering of software will be increasingly fundamental to achieving our war fighting vision of tomorrow. The future vision for software within the Air Force must focus on integration of software as a part of our system engineering disciplines, changing the culture of our software professionals

and managers, and focus on software components as a part of our larger information enterprise. Focus on these areas will ensure that software, and our software professionals, will be able to deliver advanced military-use capabilities of unmatched quality with accelerated delivery timeframes. ♦

Reference

1. Brooks, Frederick P. The Mythical Man-Month: Essays on Software Engineering. 1st ed. New York: Addison-Wesley, 2 Aug. 1995.

Notes

1. Joint Vision 2020 <www.dtic.mil/jointvision>.
2. Center for Systems Engineering <<http://cse.afit.edu>>.
3. To see the GCSS-AF Integrated Framework in action, Air Force members can access the Air Force Portal at <<https://www.my.af.mil>>. For additional articles on GCSS-AF, please see the Aug. 2003 edition of CROSSTALK.

About the Author



John M. Gilligan is the Air Force chief information officer and the principal advisor to the Air Force leadership on information management, business processes, and information technology standards. He leads the Air Force in creating and enforcing information technology (IT) standards, and promoting and shaping an effective strategic and operational IT planning process. Gilligan also leads the Air Force in acquiring IT systems, ensuring that the conduct of IT processes is timely; cost-effective; follows all applicable statutes, regulations, and policies; and provides the best available capability consistent with requirements and within available budget resources. Prior to joining the staff of the secretary of the Air Force,

Gilligan served as the CIO for the Department of Energy where he developed and directed the IT management strategies, policies, and practices for the department. His awards include the Meritorious Civilian Service Medal, the Presidential Meritorious Executive Rank Award, and the Presidential Distinguished Executive Rank Award. He has a bachelor's degree in mathematics from Duquesne University, a Master of Science in computer engineering from Case Western Reserve University, and a Master of Business Administration from Virginia Polytechnic Institute and State University.

AF-CIO

1155 Air Force Pentagon

Washington, D.C. 20330-1155

E-mail: afcio@pentagon.af.mil

WEB SITES

The United States Department of Defense

www.dod.gov

The Department of Defense's (DoD) mission is to provide the military forces needed information to protect the security of the United States. The department's headquarters is at the Pentagon. DefenseLINK is the official Web site for the DoD and the starting point for finding U.S. military information online. Its information includes official, timely, and accurate information about defense policies, organizations, functions, and operations for military members, DoD civilians, military family members, the American public, the Congress, and the news media. DefenseLINK also hosts the DoD Resource locator, a part of the Government Information Locator Service that is intended to help citizens identify, locate, and retrieve information about their government.

The United States Army

www.army.mil

This official Web site for the U.S. Army provides updated news announcements and reports, information on leaders, career management, well-being, references, an extensive site index, and more. Information quick links include Army leadership, Army Reserves, Army National Guard, public affairs, recruiting, retirees, and more. Links are also provided to U.S. Army publications and other services' news.

The United States Navy

www.navy.mil

This is the official Web site for the U.S. Navy, provided by the Navy's Office of Information, Washington, D.C., in cooperation with the Space and Naval Warfare Systems Center Charleston, Pensacola, Fla. The site provides information and news about the U.S. Navy to the Navy community, U.S. citizens, and the media. Information on the site is updated daily, and includes a comprehensive alphabetical index.

The United States Marine Corps

www.usmc.mil

This is the U.S. Marine Corps official Web site where you will find quick links to New From the Front, 2003 Concepts and Programs, the Marine Corps Institute, general officer biographies, Marine Band, and more. Information on the site is directed toward the Marines community, U.S. citizens, and the media. News is updated regularly and includes a photo gallery, press releases, and video archive.

The United States Air Force

www.af.mil

This is the official Web site of the U.S. Air Force. It provides news and information about the U.S. Air Force to the Air Force community, the media, U.S. citizens, and more. News is updated frequently and links are provided to the Air and Space Expeditionary Force Center, the Air Force Media Center, the Air Force Operations Center, Airman magazine, senior leadership, and more.

Defense Information Systems Agency

www.disa.mil

The Defense Information Systems Agency (DISA) is a combat support agency responsible for planning, developing, fielding, operating, and supporting command, control, communications, and information systems that serve the needs of the Department of Defense (DoD) and other government offices. DISA is a provider of integrated information solutions to DoD and non-DoD customers. Today, DISA is in the process of consolidating computer services. By September 2005, DISA computing services will consist of one headquarters component, four production systems management centers, and several optimally staffed processing sites, reducing the workforce by 1,200. DISA has been awarded five Joint Unit Meritorious Service Awards.

Charting the Course for the Department of the Navy's IM/IT Transformation

David M. Wennergren
Department of the Navy



The Department of the Navy's chief information officer talks about the department's information management/information technology transformation agenda and what the Navy Marine Corps Team is doing to answer the challenge to "deliver tomorrow, today."

As the Department of the Navy (DON) moves forward to meet the challenges of the 21st century, we are embarking on a journey of information management/information technology (IM/IT) transformation that will usher in new ways of deterring conflict, new capabilities for waging war, and new technologies that will lead to major increases in operational effectiveness. This transformation will serve as the foundation for a network-centric environment and knowledge dominance, and provide next-generation capabilities to the naval warfighting team.

Key to developing an effective, executable IM/IT transformation strategy is having the right leadership team. In recent months, we have been engaged in a significant effort to restructure IM/IT governance across the DON. This restructuring has helped to strengthen, align, and integrate our IM/IT efforts across the Navy and Marine Corps and to ensure department-wide alignment of IM/IT efforts with warfighter priorities.

A key element of the restructuring was the designation of the Director for Space, Information Warfare, Command and Control Rear Adm. Thomas E. Zelibor to be dual-hatted as the DON deputy chief information officer (Navy); and the Director for Command, Control, Communications, and Computers Brig. Gen. John R. Thomas to be dual-hatted as the DON deputy chief information officer (Marine Corps). The formalization of what had previously been ad-hoc relationships has significantly enhanced and better aligned the way we manage IM/IT across the Navy Marine Corps Team. It also ensures that we have an integrated vision and strategy and aligned execution.

The DON IM/IT leadership team has recently published the DON IM/IT Strategic Plan for 2004-2005. It is also actively engaged in developing the DON IM/IT Enterprise Implementation Plan, which will link vision and strategy to programmatic and budgeting guidance and serve as the basis for approving and funding future IM/IT investments. Our strate-

gic plan is available at <www.doncio.navy.mil>; I encourage you to read it for more detail and to see some of the success stories that demonstrate progress toward our goals.

The remainder of this article will outline the DON's IM/IT transformation agenda. It is comprised of a number of key initiatives woven together to deliver three IM/IT enterprise capabilities: a blueprint for modernization, knowledge avail-

"We have made great progress during the last year in legacy application rationalization. From our initial count of 100,000 applications, we are now down to approximately 5,000 and are on our way to a target goal of 2,000 applications."

ability and dominance, and effective resource management. All of these rest on the supporting foundations of Full Dimensional Protection and an exceptional IM/IT work force.

A Blueprint for Modernization

The capability to deliver a standards-based enterprise architecture and successful network integration strategy are essential elements of the DON IM/IT blueprint for modernization. Network-centric operations and warfare require that we take an enterprise approach toward management of our information assets. This reality drove the decision to pursue the innovative outsourcing strategy that is the Navy Marine Corps Intranet (NMCI).

The NMCI provides the full range of

state-of-the-art network-based information services through a performance-based contract using state-of-the-market equipment and leading industry service providers. It replaces numerous independent and disparate networks ashore with a single secure network and is a vital part of the Department of Defense (DoD) Global Information Grid, interfacing with the Navy afloat network and the Marine Corps tactical data network to enhance the flow of critical information to forward-deployed forces.

The NMCI significantly improves the security of our IT enterprise; enables greater sharing of knowledge and improved interoperability; and gives the Navy and Marine Corps secure, universal access to integrated voice, video, and data communications. The NMCI is a major first step in the DON's strategy to develop and maintain a single, seamless, secure network – our platform for network-centric warfare and knowledge dominance.

The NMCI is the foundation for much of the IM/IT transformation that is going on throughout the DON. It provided the opportunity for the DON to gain visibility of all its shore-based information assets: hardware, software, and data. The NMCI became a forcing function to standardize across the Navy and Marine Corps and reduce the number of legacy applications in use. We have made great progress during the last year in legacy application rationalization. From our initial count of 100,000 applications, we are now down to approximately 5,000 and are on our way to a target goal of about 2,000 applications. Additionally, perennial challenges of real-time collaboration, enterprise-wide applications, and authoritative data sources can be met through the NMCI.

Knowledge Dominance

The two core themes of the Navy Marine Corps IT team are network-centric operations and knowledge dominance for the Naval warfighting team. Knowledge dominance is essential – having access to the right information at the right time from

authoritative data sources to allow quicker decision making and collaboration. As we achieve a seamless enterprise network structure, we are simultaneously transforming the way information is shared to truly achieve knowledge dominance. Commands across the Navy Marine Corps team are leveraging the tenets of knowledge management and using Web service solutions to create virtual collaboration environments.

Operational forces recognize the power of collaboration and knowledge sharing and have become champions for knowledge management. A highly effective knowledge-management pilot program is ongoing at Commander, Submarine Group 10 in Kings Bay, Ga. It involves the Trident submarine *blue* and *gold* crews; as a crew comes off deployment to shore, they can still maintain their proficiencies and share and collaborate during that off-cycle time.

At Commander, Naval Reserve Force, there is a project to reengineer the entire claimancy using knowledge management as the foundation for that transformation. Recognizing the power of knowledge management throughout the Navy Marine Corps team, Commander, Network Warfare Command Vice Adm. Dick Mayo is leading a flag officer level knowledge-management steering group to make sure we continue to embrace and deploy knowledge management solutions for the warfighter.

The Navy Marine Corps Portal (NMCP) will provide an integrated, collaborative environment and personalized, role-tailored access of information in real time. This single integrated portal structure will allow our organizations to focus on content delivery, and avoid the costs of individually developing portal features and functions. Authoritative data sources and enterprise-wide applications will be accessed through the portal and secured via Public Key Infrastructure (PKI) facilitated by the Common Access Card (CAC) and e-authentication. The NMCP will reduce application costs and improve information security, providing our Sailors, Marines, and civilians with access to the intellectual capital of the entire Navy Marine Corps team.

Effective Resource Management

The IT investment and expenditure decisions made by Navy and Marine Corps commands must be aligned with our strategic goals, and must improve combat capability, warfighting readiness, and mis-

sion performance. To this end, we are engaged in developing a capital planning process that validates IM/IT requirements as part of the Program Objectives Memorandum/budget process, and an Enterprise Implementation Plan that links our long-term vision and strategy to programmatic and budget guidance.

Our Capital Planning Integrated Process Team is working to combine classical capital planning methods with elements of the Marine Corps' Information Technology Steering Committee process that answers the needs of the entire department. Though producing the Enterprise Implementation Plan will be a complex task, we have taken an important step in the right direction. Our "Information Technology Policy Guidance for Fiscal Year 2004 Expenditures" memo-

"Each of us must be a change leader. Each of us must be willing to do our part to leverage technology as part of a larger effort to reinvent and reinvigorate our warfighting processes."

randum to Navy Echelon II and Marine Corps Major Commands is an executive summary of national, DoD, and DON policies in place to promote secure, interoperable, and standards-based IT solutions. It is referenced in the appropriate program authorization documents, requiring comptrollers to ensure compliance before approving disbursements for IT.

Full Dimensional Protection

The 21st century presents new challenges for continued maritime dominance and national security. We have crafted an approach we call full dimensional protection. Joint Vision 2020 states that *full dimensional protection* is achieved "through the tailored selection and application of multi-layered active and passive measures." For the DON, that protection takes three forms: (1) protecting knowledge pathways through information assurance and defense in depth, (2) protecting our centers of knowledge through critical infrastructure protection, and (3) protecting our knowledge workers through

efforts to protect individual privacy.

Information assurance is required in the DON to protect and defend information and information systems by ensuring their availability, integrity, authentication, confidentiality, and non-repudiation. We have adopted the defense-in-depth strategy to mitigate the risk associated with a single point of failure.

The Common Access Card (CAC) is the cyber and physical identification card for all DoD active duty military, reserve military, civilian, and eligible contractor personnel. More than 3 million CACs have been distributed to DoD personnel (1.1 million within DON). The CAC, with its DoD PKI credentials, is the cornerstone for improving the department's information assurance posture by facilitating secure access to physical and cyberspaces.

The DON Critical Infrastructure Protection program ensures protection of both cyber and physical mission-essential infrastructures, employing the Naval Integrated Vulnerability Assessment process to identify single points of failure of critical assets. Local, state, and DoD leaders have enthusiastically endorsed this process. We have also developed a Consequence Management Planning Guide to assist military commanders in the development of continuity of operations plans, and are working with state and local governments to share lessons learned.

Privacy is the third leg of our Full Dimensional Protection program. Now, more than ever, striking the delicate balance between personal privacy and national security is a challenge faced by the entire nation. The DON recognizes this fact and is taking steps to ensure privacy protection through tools and policies to aid in the protection of personal information in DON systems.

IM/IT Work Force: The Foundation for Future Success

Foundational to the DON's transformation is our work force. The DON's current successes and future ability to transform the IM/IT enterprise are directly attributable to the outstanding men and women of our military and civilian service.

We have an extremely intelligent and innovative work force, and our IT professionals are up to the challenges of the 21st century digital revolution. However, in our rapidly changing world, the skill sets and knowledge required of our IM/IT work force are also changing rapidly. For this reason, we have provided

tools to help our work force assess their current and future needs and develop competencies related to those needs.

The entire federal government has now adopted these career-planning tools. Our leveraging of such tools and continuous learning programs will help us develop and retain the skilled work force needed to continue our IM/IT transformation.

Leading Change

This is a time of great change – viewed with consternation by some, but fortunately embraced by many more as a time of great opportunity. I think the measure of an effective chief information officer is the ability to lead change. You have to understand technology, but technology is only part of the answer. I spend a far greater part of my day working the cultural change issues that go along with making an organization transform.

Each of us must be a change leader. Each of us must be willing to do our part to leverage technology as a part of a larger effort to reinvent and reinvigorate our warfighting processes. At the recent Navy

Information Professional Community Summit, the chief of Naval Operations asked a group of IT professionals “to deliver tomorrow, today.” The combination of a need to understand and embrace the future, but to deliver results now, is

right on target. Choosing to change means accepting risks. Choosing not to change, in today’s world, risks irrelevancy. I am honored to be a part of an outstanding Navy and Marine Corps team that has chosen to champion change. ♦

About the Author



David M. Wennergren is Department of the Navy chief information officer (DON CIO) reporting directly to the secretary of the Navy.

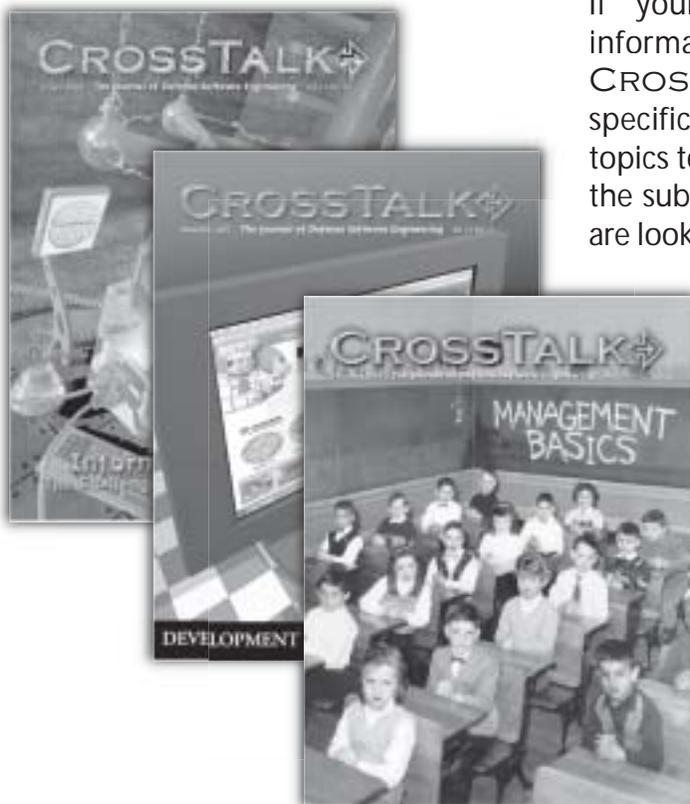
He provides top-level advocacy in the development and use of information management/information technology (IM/IT) and creation of a unified IM/IT vision for the Navy-Marine Corps team. He is also the chair of the Department of Defense Smart Card Senior Coordinating Group, where he is responsible for the deployment of smart card technology, including the Common Access Card. He has received both the DON Superior and Meritorious Civilian

Service Awards, and the Office of the Secretary of Defense Exceptional Civilian Service Award. Wennergren has a bachelor’s of arts degree in communications/public relations from Mansfield State University. He was a recipient of a Secretary of the Navy Civilian Fellowship in Financial Management, culminating in a master’s degree in public sector financial management from the University of Maryland’s School of Public Affairs.

Point of Contact:

Lynne D. Gaudreau
Phone: (703) 602-6705
Fax: (703) 601-2182
E-mail: lynne.gaudreau@navy.mil

CALL FOR ARTICLES



If your experience or research has produced information that could be useful to others, CROSSTALK can get the word out. We are specifically looking for articles on software-related topics to supplement upcoming theme issues. Below is the submittal schedule for three areas of emphasis we are looking for:

System Assessments and Certifications

June 2004

Submission Deadline: January 19, 2004

Best Practices

Open

Submission Deadline: Rolling

Open Forum

Open

Submission Deadline: Rolling

Please follow the Author Guidelines for CROSSTALK, available on the Internet at: <www.stsc.hill.af.mil/crosstalk>. We accept article submissions on all software-related topics at any time, along with Letters to the Editor and BackTalk.

The Fire Support Software Engineering Division Achieves CMMI Level 5

Milton Smith

Communications-Electronics Command Software Engineering Center Fire Support Software Engineering

Phil Sperling

Telos-OK



The U.S. Army Communications-Electronics Command (CECOM) Software Engineering Center (SEC) Fire Support Software Engineering (FSSE) Division and its prime contractor, Telos-OK, have attained the Capability Maturity Model Integration (CMMI®) Level 5 rating in systems and software engineering. After 13 years of continuous pursuit and practice of a software and systems improvement program, the CECOM SEC FSSE has become the first Department of Defense organization to attain CMMI Level 5.

The U.S. Army Communications-Electronics Command (CECOM) Software Engineering Center (SEC) Fire Support Software Engineering (FSSE) Division and its prime contractor, Telos-OK, achieved an unprecedented milestone on Aug. 6, 2003. The appraisal team leader from the Software Engineering Institute (SEI) announced that CECOM SEC FSSE had achieved Capability Maturity Model Integration (CMMI®) Level 5. They are the first Department of Defense (DoD) agency to achieve this rating, and only the second organization within the United States to do so.

In August 1990, the U.S. Army's Army Material Command (AMC) began an affiliation with the SEI. The purpose of this affiliation was to generate methods to continually improve the software engineering capability of all U.S. Army software development agencies.

The SEI introduced the AMC to the Capability Maturity Model® for Software (SW-CMM®), which provides a model of software engineering key practices accepted by the software industry. The AMC selected the CECOM SEC to take the lead in implementing the methods associated with this model. The SEC subsequently selected the FSSE Division, which is located at Fort Sill, Okla., to initiate formal software engineering process improvement in accordance with the guidelines set forth in the CMM.

The team of the SEC FSSE and Telos-OK has been a partnered organization since the founding of the FSSE by the predecessor to the AMC in 1976. This partnership has survived and thrived through six full and open five-year contract competitions during the intervening 26-year period. The processes are owned by the government, and practiced, controlled, and changed by the government and contractor team.

The FSSE began its software engi-

neering process improvement program by establishing a benchmark activity to determine its current state of software engineering maturity, as described in the

"This was the first time that any organization affiliated with the SEI had moved from Level 1 to Level 3 without an interim appraisal of Level 2. The Level 3 placed the FSSE in the top 15 percent of software development organizations around the world."

CMM for Maturity Levels 1 through 5. This appraisal was conducted in February 1991, and identified the organization as a CMM Level 1 with a large portion of the Level 2 key practices in place. Members of the FSSE, the SEC Ft. Monmouth, the Mitre Corp., and the SEI conducted this appraisal.

Subsequent to this first appraisal, the FSSE immediately developed and implemented a plan for process improvement, which primarily focused upon key practices associated with organization and system-level process documentation, personnel training, managing process improvement, and technology innovation. The implementation of this plan was completed in September 1994, and

the FSSE conducted its second benchmarking activity. Members of the FSSE, the SEC Ft. Monmouth with full participation of representatives from Mitre, and the SEI conducted the appraisal.

The actual appraisal methodology had become significantly more rigorous over the past three years; however, the organization was appraised at a solid CMM Level 3. This was the first time that any organization affiliated with the SEI had moved from Level 1 to Level 3 without an interim appraisal of Level 2. The Level 3 placed the FSSE in the top 15 percent of software development organizations around the world.

Level 4: The Next Step

Following this second appraisal, the FSSE refined its process improvement plan to move next to CMM Level 4. The refinements to this plan focused primarily on establishing a viable management-through-measurement program. The key attribute of a CMM Level 4 organization is its ability to quantitatively control the process performance and product quality of its software development efforts. However, there were no other Level 4 organizations in the world from which to draw lessons learned for moving from a Level 3 to a Level 4 process maturity level. It took quite a bit of study to gain an adequate understanding of what the CMM actually intended with its Level 4 key practices.

We turned to Lockheed-Martin, a CMM Level 5 organization, and the SEI to assist in deciphering the Level 4 key practices. The SEI and Lockheed-Martin provided excellent guidance and assistance in identifying methods and means for the FSSE to adequately plan for the Level 4 process changes. The following bullets elaborate upon some of the major implementations identified for the move to Level 4:

- A formal methodology was documented that described how the

organization collected, displayed, and analyzed software engineering data.

- A new tool was created to provide real-time visibility to managers and developers into the progress, process performance, and product quality of all software engineering efforts.
- Training in statistical process control (SPC) was provided to managers and engineers across the organization. Subsequently, SPC was tagged to those critical processes used for software development.
- Additional reviews and reporting methodologies were employed within the organization's standard software process.

The implementation of the process improvement plan was completed in November 1997, and the third benchmarking activity was conducted. During this appraisal, the FSSE was identified as meeting the criteria for CMM Level 4. Members of the FSSE, Lockheed-Martin, and the SEI conducted this appraisal. This newest rating placed the FSSE in the top 2 percent of software development agencies around the world.

The FSSE then implemented plans to improve its Level 4 process and advance toward CMM Level 5. During this important process upgrade, the SEI, the DoD, and the software community established the CMMI project, and a new model was released in the latter part of 2001. The accompanying, and more rigorous, Standard CMMI Appraisal Method for Process Improvement (SCAMPISM) was released in early 2002.

This new model provides for an integration of software and systems engineering. Level 5 means that process improvement is a way of life for an organization. Improvements flow naturally from the work force, and through the exercise of the organization's development processes. The key attributes of a Level 5 organization are predictability, efficiency, and high quality. The FSSE adjusted its improvement program to adopt the new model, as well as the new SCAMPI appraisal methodology.

The latest appraisal was concluded on Aug. 6, 2003 with the achievement of CMMI Level 5. A notable aspect of this newest appraisal method was the length of time needed to complete the event. The appraisal, which was led by members of the SEI, actually began with a mini-appraisal on March 12, 2002. The organization conducted three mini-appraisals from March 2002 through June 2003.

These mini-appraisals were used as a means to facilitate the gathering, review, and rating of a tremendous amount of documented evidence. This mini-appraisal technique was found conducive to the increased rigor of the SCAMPI method, and allowed the appraisal team to prepare and conduct a smooth one-week on-site interview period.

The achievement of this benchmark exemplifies the seriousness the FSSE holds for developing and maintaining mission-critical, highly reliable, tactical software used in support of the nation's fire support warfighters. Fire support is the collective and coordinated use of indirect fires, target acquisition data, armed aircraft, and other lethal and non-lethal means against ground targets in support of the maneuver force operations.

"The achievement of this benchmark exemplifies the seriousness the FSSE holds for developing and maintaining mission-critical, highly reliable, tactical software used in support of the nation's fire support warfighters."

Characteristics of the Level 5 Process

CMMI Level 5 requires advanced project management, which gives increased visibility into and control of the software and system engineering process. The following traits are indicative of this level of management:

- Evidence of decisions based upon *quantitative* analysis.
- Evidence of *formal* decision analysis.
- Evidence of identifying *root causes* of problems – not just software defects.
- Evidence of *measuring* improvements against projected outcomes. These are process and technology improvements.
- Evidence of *institutionalization* across all projects.

Most importantly, a culture exists that invites ingenuity and creativity from the work force. This culture is evidenced through the documented involvement of the work force in process improvement.

Missions Supported by the Level 5 Process

The FSSE was among the first Life-Cycle Software Engineering Centers established to support mission critical software for the U.S. Army. The center performs diverse work in all areas of the fire support domain to include command and control, target acquisition, tactical fire control, and technical fire control. Supporting work includes fire support automated testing, validation (regression, ballistics, stress, and interoperability, using a mix of tactical equipment and simulation), systems integration, system emulation, or porting and training. The knowledge base required to accomplish the mission includes system and software engineering, NATO, joint and Army interoperability, software training and fielding, doctrine, tactics, radar, cannon, missile, auxiliary equipment, and embedded systems.

Within the joint service community, the Joint Variable Message Format Bit-Oriented Message (BOM) standard used between services for interoperability was base-lined using a core set of 63 Army Variable Message Format BOM that were invented, developed, and matured by the FSSE. This advanced messaging capability is a direct result of the processes that were used in the FSSE's system-of-systems package development and maintenance methodology.

The FSSE has successfully fielded 73 major fire support weapon system software versions. We have transitioned eight weapon systems, developed 20 new fire support weapon systems, performed 68 major weapon systems upgrades, developed 27 major programming support and automated test systems, and developed 18 major prototype systems. Currently, we are responsible for more than 9.5 million source lines of code (SLOC). This includes over 600,000 in-house developed Ada SLOC for new systems and 2,197,000 SLOC transitioned and updated or reused in new systems.

The FSSE has supported the Army's transformation through Task Force XXI and the current First Digitized Division. The FSSE accomplished this support by providing more updated systems (10 systems in all) to Task Force XXI and the First Digitized Division than any other

SM SCAMPI is a service mark of Carnegie Mellon University.

government agency or contractor. The FSSE performed the additional task without impacting on-going Army operational needs, doing it under cost and on schedule.

Return on Investment

Since the advent of the original AMC initiative for process improvement, thousands of organizations and projects around the world have instituted formal software engineering process improvement programs. The FSSE continues to be at the forefront in industry in process improvement, leveraging the CMMI Level 5 processes to develop and maintain mission critical, high reliability, command and control, real-time acquisition fire control and fire direction tactical systems. A significant return on investment has been realized through increased product quality (reduced defects) and increased productivity (decreased cost to the customer). The following details some of these returns:

- **System Growth.** The organization has gone from maintaining over 1.5 million LOC in 1993, to over 9.5 million LOC today. This has been accomplished without any increase in staffing. As the process has matured, process efficiencies and new technology have provided the increased maintenance ability. One of these process efficiencies is advanced reuse of components.
- **Size Stability.** The organization's ability to accurately predict the size of the projects at the beginning of development has improved 250 percent. As most companies realize, this ability is critical in estimating staffing and other resource needs.
- **Software Quality.** Defects found during formal testing have decreased by over 80 percent.
 - Defects found on previous version = 2,881; LOC = 2,050,739 for a rate of 1.40 defects per KLOC.
 - Defects found on most recent version = 457; LOC = 1,966,702 for a rate of 0.23 defects per KLOC.
 - Decrease in defect rate = 83.57 percent.

This provides engineers time to do valuable follow-on efforts, instead of concentrating on rework and retest.

- **Productivity.** The organization's LOC/hour rate has increased by 48 percent. This is even more significant when considering the increased quality.

- **Cost.** For the 13-year period encompassing the FSSE's current process improvement efforts, the average inflated 13-year operating cost would have been approximately \$59 million per year. Because of continuous process improvements, the average inflated cost per year is approximately \$30 million. This represents significant savings attributable to process improvement efforts.

Specific Improvements

We have seen in similar reports a lack of specific improvements identified, which

“For the 13-year period encompassing the FSSE's current process improvement efforts, the average inflated 13-year operating cost would have been approximately \$59 million per year. Because of continuous process improvements, the average inflated cost per year is approximately \$30 million.”

were actual implementations through the maturing of the process. Here are several that have been implemented here:

- **Improved Software Development Model.** The organization has implemented the Incremental Software Development process, which provides for finding defects earlier in development, as well as better handling of requirements and requirements changes. For some smaller projects, or as needed, the process still allows for use of the Waterfall Model.
- **Advanced Reuse.** As mentioned earlier, maintaining a reuse repository, administering its use, and wiring it

into the process has provided significant gains in efficiency and quality. This advanced reuse capability is fully supported by the processes and is made possible because the FSSE has software responsibility for numerous related domain-specific systems.

- **Automated Tools.** The organization has developed automated tools that provide significant improvements in project management, software development, and testing. Some of these are the following:
 - **TRACKER:** A locally developed tool used to manage the contract dependencies between the government customer and its support contractors.
 - **CMDB:** A locally developed tool used to control the creation of software systems and their components. This tool has numerous features in one product, which can only be found through multiple products on the market.
 - **ABTCS and TSS:** Locally developed tools that provide fast and thorough means for testing software and validating a system's software baseline capabilities as the system's software continues to evolve.
- **Formal Inspections.** Although implemented years ago, the organization continues to refine this process, and looks deeper into the various products to identify and eliminate errors. These inspections begin during requirements development and go through the testing and training products.
- **Numerous Development Platforms.** The organization has moved from numerous program support environments (PSEs) into a single PSE, thereby reducing procedural and technical training burdens associated with the PSE and allowing engineers and programmers to move quickly and easily between systems as needed. In addition, we pursue and continue to move toward standard unified tactical platforms for all pertinent systems. This allows multiple tactical systems to use a common hardware platform and to reduce the maintenance cost of maintaining the hardware through volume usage and use of parts of common items.
- **Open Systems Operating System.** The FSSE is one of the first to move all tactical systems from vendor-specific operating system solutions to a

solution based on a free and open system operating system kernel (i.e. Linux). This has relieved the FSSE from being held hostage to external vendors and their market-driven desires, and provided uniform processes, tools, and engineering methodologies to be applied across the multiple systems that the FSSE is responsible for maintaining and enhancing for the Army and Marine Corps.

- **User Interface.** The organization has found that one of the most significant advances came through the establishment of local and worldwide interface/interoperability boards. These groups provide for the direct involvement of user representatives in the details and decisions throughout software and system development. These groups are proving extremely critical in a world that is moving more toward common interfaces.

Conclusion

The CECOM SEC FSSE now looks forward to sharing its accomplishments and experience with others within the federal government, DoD, and the software industry. There are numerous areas where this organization could provide benefit to other agencies:

- Technical assistance in the primary areas of software development (configuration management, quality assurance, testing, engineering, etc.).
- Tool development and maintenance.
- Formal inspections.
- Process improvement guidance (Level 2 to Level 5).
- Formal appraisal conduct and assistance.

It is realized that CMMI Level 5 is not an end but simply another step in the evolution of the software process. Establishing this benchmark provides the organization the ability to see the promise for the future. ♦

About the Authors



Milton Smith is the acting chief of the Communications-Electronics Command Software Engineering Center Fire Support Software Engineering (FSSE) Division located at Ft. Sill, Okla. Prior to his current assignment, he was the senior staff software specialist to the FSSE chief, and before that a branch manager of the Software Engineering Branch for all the FSSE tactical system. Smith has also worked for the Sperry Univac Corp., and for the United States Field Artillery School as an instructor in the advanced technology branch of the Communications Electronics Division. He wrote the course material for the fundamentals of microcomputers, operation, and maintenance used within the school.

E-mail: smithmb@fssec.army.mil



Phil Sperling is principal process engineer for Telos-OK, a software development contractor for the U.S. Army. He has successfully guided his organization through several stages of process improvement using the Capability Maturity Model® practices and the Capability Maturity Model Integration (CMMI®). Most recently, his organization achieved CMMI Level 5 using the System Engineering/Software Engineering Staged Version 1.1. Sperling retired from the U.S. Army in 1992. He served as a software-engineering officer and was deployed to support Intelligence and Electronic Warfare, Communications, and Field Artillery Systems during Operation Desert Shield and Operation Desert Storm. He also served a combat tour in Southwest Asia, and held leadership positions in Armor and Field Artillery. Sperling has a bachelor's degree in business from the University of Arkansas and a master's degree in business administration from Cameron University.

E-mail: phillip.sperling@us.army.mil



Get Your Free Subscription

Fill out and send us this form.

OO-ALC/MASE

6022 FIR AVE.

BLDG. 1238

HILL AFB, UT 84056-5820

FAX: (801) 777-8069 DSN: 777-8069

PHONE: (801) 775-5555 DSN: 775-5555

Or request online at www.stsc.hill.af.mil

NAME: _____

RANK/GRADE: _____

POSITION/TITLE: _____

ORGANIZATION: _____

ADDRESS: _____

BASE/CITY: _____

STATE: _____ ZIP: _____

PHONE: (____) _____

FAX: (____) _____

E-MAIL: _____

CHECK BOX(ES) TO REQUEST BACK ISSUES:

- NOV2002 PUBLISHER'S CHOICE
- DEC2002 YEAR OF ENG. AND SCI.
- JAN2003 BACK TO BASICS
- FEB2003 PROGRAMMING LANGUAGES
- MAR2003 QUALITY IN SOFTWARE
- APR2003 THE PEOPLE VARIABLE
- MAY2003 STRATEGIES AND TECH.
- JUNE2003 COMM. & MIL. APPS. MEET
- JULY2003 TOP 5 PROJECTS
- AUG2003 NETWORK-CENTRIC ARCHT.
- SEPT2003 DEFECT MANAGEMENT
- OCT2003 INFORMATION SHARING
- NOV2003 DEV. OF REAL-TIME SW
- DEC2003 MANAGEMENT BASICS

TO REQUEST BACK ISSUES ON TOPICS NOT LISTED ABOVE, PLEASE CONTACT KAREN RASMUSSEN AT <[KAREN.RASMUSSEN@HILL.AF.MIL](mailto:karen.rasmussen@hill.af.mil)>.



The AV-8B Team Learns Synergy of EVM and TSP Accelerates Software Process Improvement

Lisa Pracchia
Naval Air Systems Command

This article is a continuation of the success story published in the September 2002 issue of CROSSTALK titled "AV-8B's Experience Using TSP to Accelerate SW-CMM Adoption" [1]. The original article shared AV-8B's lessons learned in achieving the Capability Maturity Model® for Software (SW-CMM®) Level 2 maturity in just 14 months. This article continues where that previous article left off. It explains the accelerating initiatives the AV-8B used to achieve SW-CMM Level 4 maturity in just another 16 months instead of the 50-month average reported by the Software Engineering Institute.

It is a business reality that most software projects are significantly behind schedule or never reach completion. According to a recent Standish Group Chaos Study [2], only 28 percent of all software projects finish on schedule, within budget, and contain all the features/functions originally specified.

This article describes how the Naval Air Systems Command's (NAVAIR) AV-8B Joint System Support Activity (JSSA) overcame those challenges by developing a strong process infrastructure based on two synergistic process improvement initiatives. Through those initiatives they surpassed their goals by reducing schedule variance by 90 percent and still achieved the measurable benefits of a Capability Maturity Model® for Software (SW-CMM®) Level 4 organization 60 percent faster than the average organization¹.

"The recipe for accelerating AV-8B's climb up the software maturity ladder and realizing the related benefits," says the AV-8B JSSA's leader Dwayne Heinsma, "cen-

tered around identifying champions and using process discipline as an enabler."

Those champions included the following:

- A Personal Software ProcessSM (PSPSM)/Team Software ProcessSM (TSPSM) champion leading the software team.

- Most importantly, it took teamwork.

Setting the Foundations

The AV-8B integrates new capabilities into the Harrier aircraft for the U.S. Marine Corps and its allies, Spain and Italy. Like many other organizations, its primary process improvement goals are to reduce cycle time and increase quality. To help achieve these goals, the AV-8B implemented two complimentary process improvement initiatives – EVM and PSP/TSP.

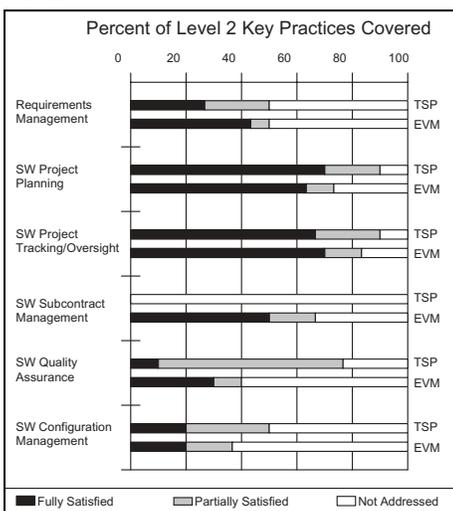
EVM is a management technique that integrates cost, schedule, and technical performance. The AV-8B began its EVM journey in 1998. By the end of 2001, the AV-8B had successfully certified their EVM system based on the Department of Defense's stringent 32-point criteria². Capability mileposts along that road included documenting organizational standard processes for activities such as negotiating commitments; estimating, planning, and tracking all project work based on a standard work breakdown structure; assigning and communicating responsibilities; managing critical paths and resourced dependencies within and across projects; and taking corrective actions based on established thresholds.

The second significant process improvement initiative was the AV-8B's adoption of the TSP as its standard software process. The TSP is a high-maturity process for software teams developed by the Software Engineering Institute [3]. The AV-8B launched its first TSP new-development project at the beginning of 2001 followed by a second TSP maintenance project in mid-2002.

The TSP provided the software project teams a complete package of training, tools, processes, coaching, and mentoring.

"In short, TSP was the singular reason why the AV-8B achieved a Level 4 rating in record time. The AV-8B's TSP implementation nearly fully satisfied all Level 4 key practices – all without needing a separate quantitative management plan or separate measurement group."

Figure 1: The AV-8B Analysis of How EVM and TSP Satisfied Level 2 Key Practice Areas



- An organizational process champion leading the development and the institutionalization of organizational standards.

- Senior managers championing the overall effort and removing roadblocks (establishing PSP/TSP as well as Earned Value Management [EVM] as the standard way of doing business at the JSSA).

SM Personal Software Process, Team Software Process, PSP, and TSP are service marks of Carnegie Mellon University.

From day one, these teams had a customizable framework with which to estimate, plan, track, communicate, and measure the quality of their software processes and work products. In addition, standard TSP roles established within each software team the responsibilities for communicating and coordinating software team activities with the larger AV-8B organization.

Measuring EVM/TSP Impact

In September 2002, the AV-8B conducted a SW-CMM Level 3/Level 4 CMM-based Appraisal for Internal Process Improvement (CBA-IPI). The assessment team had a secondary objective of analyzing the benefits of EVM and TSP on the AV-8B's software process maturity. To accomplish their objective, the assessment team flagged observations during the assessment that mentioned either EVM or TSP.

At the conclusion of the CBA-IPI, AV-8B's System Software Engineering Process Group (SSEPG) lead took those flagged observations and mapped each one back to a specific SW-CMM key practice it supported. Then, using her SW-CMM experience and professional judgment, the SSEPG lead independently determined which key practices were fully satisfied, partially satisfied (i.e., additional effort was needed), or were not at all satisfied by each EVM or TSP observation.

The next three sections illustrate the results of this analysis summarized at the key process area (KPA) level.

Impact on Level 2 Key Practices

The focus of SW-CMM Level 2 is on basic management processes. The AV-8B's EVM and TSP implementations satisfied the majority of Software Project Planning and Software Project Tracking and Oversight key practices, as illustrated in Figure 1. EVM and TSP also partially satisfied many other Level 2 key practices.

EVM satisfied the intent for all facets of a project through an institutionalized system of agreed-upon commitments, well-defined plans, documented methods for tracking actual performance against plans, procedures for making course corrections, and training to perform related tasks.

TSP provided those exact same capabilities at the software team level through a defined implementation strategy, a documented project initiation process called a *launch*, a similar replanning process called a *relaunch*, and a project status process performed weekly by TSP software teams. It is important to note that both EVM and TSP

use the earned value (EV) method of reporting progress, and that the software teams' EV was fed into the organizations EV to achieve overall project status measures.

EVM and TSP at the AV-8B only partially satisfied the remaining Level 2 KPAs. That is because neither EVM nor TSP provided specific training for these functional areas, defined functional-area activities or work products, or provided for independent quality assurance (QA) verifications. A noteworthy observation is that TSP partially satisfied the majority of software QA key practices through a TSP team role that served as a touch-point between the TSP team QA and the organizational QA. In addition, the AV-8B's EVM requirements are equally levied on its software subcontractors while TSP does not at all address subcontractor considerations.

Impact on Level 3 Key Practices

At Level 3, projects are expected to tailor a common set of documented and approved organization-wide management and engineering processes. As with Level 2, both

*“At Level 3 ...
as with Level 2, both
EVM and TSP at the
AV-8B did significantly
contribute to partially
satisfying most KPAs.”*

EVM and TSP at the AV-8B did significantly contribute to partially satisfying most KPAs. At Level 3, however, neither EVM nor TSP fully satisfied many KPAs, as illustrated in Figure 2.

The scope of TSP, by design, is limited to software team practices. This limited scope is obvious when you look at the KPAs that have a wide organizational application such as Organizational Process Focus/Definition, Training Program, and Intergroup Coordination. However, TSP fully satisfied a majority of peer review key practices through its individual review and group inspection processes. TSP also provided processes for integrating management and engineering activities to fully satisfy some Integrated Software Management key practices, as well as processes and measures to ensure quality engineering as described in the software product engineering KPA.

While EVM is organizationally focused, its processes are primarily project management related. As a result, the AV-

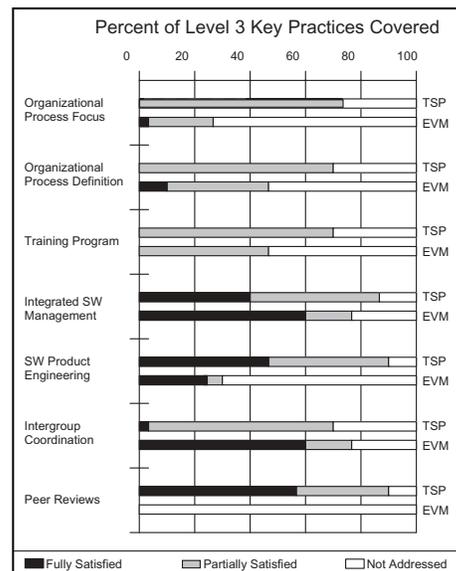


Figure 2: AV-8B Analysis of How EVM and TSP Satisfied Level 3 Key Practice Areas

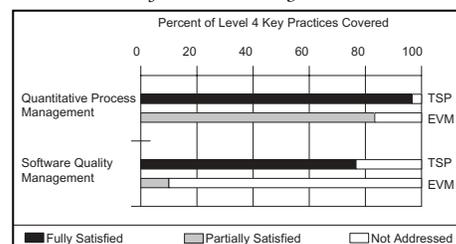
8B's EVM processes fully satisfied the majority of Integrated Software Management key practices. In addition, AV-8B's EVM guide was referenced as the organization's intergroup coordination plan. This guide provided standard processes for agreeing on commitments across teams and for identifying, tracking, and resolving intergroup issues.

Impact on Level 4 Key Practices

The focus of Level 4 is for projects to collect and use detailed measures for both process and product quality. EVM and TSP parted ways at the AV-8B at this level. In short, TSP was the singular reason why the AV-8B achieved a Level 4 rating in record time (see Figure 3). The AV-8B's TSP implementation nearly fully satisfied all Level 4 key practices – all without needing a separate quantitative management plan or separate measurement group. To the AV-8B, achieving Level 4 was not an effort but rather a natural evolution of using tools and techniques embedded in TSP.

As previously mentioned, TSP gave the software team the capability to understand and measure the quality of its software processes and work products from day one. Tracking and analyzing four basic TSP

Figure 3: AV-8B Analysis of How EVM and TSP Satisfied Level 4 Key Practice Areas



measures – size, time, defects, and completion dates – achieved this capability. In addition, TSP provided all the training, tools, and analysis procedures the software team needed to control and improve their processes using these measures. All that AV-8B needed to do in order to satisfy Level 4 KPAs was for the SEPG to fill in the organizational gaps. These gaps consisted of drafting policies, defining the organization's capability baseline, and identifying quality goals assigned to subcontractors.

EVM only indirectly supported the measurement of process quality and did not address product quality.

Realizing the Benefits

With EVM and TSP in place, and an open culture that encouraged taking qualified risks, the AV-8B rapidly enhanced its software process maturity. According to Chris Ricketts, the AV-8B lead software engineer and TSP design manager, "Success came because of the team's ability to change paradigms by abandoning the old way of doing business and implementing PSP/TSP."

AV-8B software engineer and TSP Process Manager Dave Curry says, "The software engineers showed incredible discipline in using TSP and learning how it and other artifacts applied to the various CMM concepts. We had to change what we did to develop software. We had to change how we thought about developing software. TSP is a tool – the team made it work!"

"Without a doubt, having EVM in place to monitor cost and schedule was a major contributor," adds Katie Smith, an AV-8B software quality engineer, "along with management initiative and support for process improvement."

Team culture, champions for software process improvement, sound adherence to discipline and schedule, and full management support along with focusing on EVM and TSP are the factors that made it all happen. In terms of the analysis presented in this article, EVM at the AV-8B was primarily beneficial at lower maturity levels while TSP offered both high- and low-maturity benefits. The author would further expect to see the same analysis results repeated for Level 5 KPAs that were demonstrated for Level 4 (i.e., TSP benefits take center stage while EVM bows out).

The specific benefits of EVM and TSP at the AV-8B have been significant. In 1998, before implementing EVM, the AV-8B had a schedule variance of +30 percent. Once EVM was institutionalized,

schedule variance dropped to +18 percent. TSP further reduced that variance to +2.5 percent on a product with a defect density of 2.1 defect/thousand lines of code that could not be broken in system test.

What advice does the AV-8B team have for others contemplating such an effort? Brad Hodgins, the AV-8B Software Task Team lead, says, "You should plan on having someone committed to process improvement as their primary task."

"You want to start slowly, making small changes," advises Curry. "Let people adjust and find *their* way. A team that understands that is more willing to buy in."

Also, adds Ricketts, "Don't expect this change to be easy or happen overnight. The change has to start at the management level first. Without their support it will not be successful."

As for the future, the AV-8B plans to transition process improvement to a new model, the Capability Maturity Model Integration (CMMI[®]), which integrates software engineering and systems engineering disciplines into a cohesive approach to process improvement. Heinsma is already visualizing future success for the AV-8B team. "We expect to be ready for our first formal CMMI assessment in a couple of years," he says.

With the continuing progressive teamwork evidenced by the AV-8B team, they will be ready. ♦

References

1. Hefley, B., L. Pracchia, and J. Schwalb. "AV-8B's Experiences Using the TSP to Accelerate SW-CMM Adoption." *CROSSTALK* Sept. 2002: 5-8 <www.stsc.hill.af.mil/crosstalk.2002/09/hefley.html>.
2. Johnson, J., K. Boucher, K. Connors, and J. Robinson. "Collaboration: Development & Management – Collaborating on Project Success." *Software Magazine* Feb./Mar. 2001 <www.softwaremag.com/archive/2001/feb/CollaborativeMgt.html>.
3. Humphrey, Watts S. *Introduction to the Team Software Process*. Boston: Addison-Wesley, 1999.

Notes

1. See the Software Engineering Institute's latest Maturity Profile Report at <www.sei.cmu.edu/sema/profile.html>.
2. Please see the Office of the Secretary of Defense Web site at <www.acq.osd.mil/pm> for information on earned value management and the 32-point criteria.

Additional Reading

1. Information on earned value management basics can be found at <www.acq.osd.mil/pm/evbasics.htm>.
2. Information on the Software Engineering Institute's Team Software Process can be found at <www.sei.cmu.edu/tsp>.
3. Davis, N., and J. McHale. *Relating the Team Software Process to the Capability Maturity Model for Software*. CMU/SEI-2002-TR-008. Pittsburgh: Software Engineering Institute, 2002.
4. Hefley, B., and L. Pracchia. "Accelerating SW-CMM Progress Using the TSP." Software Engineering Process Group Conference, Phoenix, AZ, Feb. 2002.
5. Crosby, L. "AV-8B JSSA Team Soars to Level 4." Naval Air Systems Command Press Release, Jan. 2003 <http://pao.navair.navy.mil/press_releases/index.cfm?fuseaction=press_release_view&Press_release_id=2016&site_id=16>.
6. Hefley, B., and L. Pracchia. "CMM + PSP/TSP → CMMI." Software Technology Conference, Salt Lake City, UT, Apr. 2003.

About the Author



Lisa Pracchia is a member of the Naval Air Systems Command's Software Resource Center. Her software background consists of process improvement, business analysis, project management, product life-cycle management, and product marketing in a wide range of industries (discrete product manufacturing, international publishing, telecommunications, and weapons systems and avionics development/support for both the government and private industry). Pracchia has a master's degree in management from the University of Redlands.

**Commander, NAWCWD
41K300D (L. Pracchia), Bldg. 1494
1 Administration Circle
China Lake, CA 93555
Phone: (760) 939-2188
DSN: 437-2188
E-mail: lisa.pracchia@navy.mil**



Object-Oriented Layers in ELIST

Mary Ann Widing, Kathy Lee Simunich, Dariusz Blachowicz, Mary Braun, and Dr. Charles Van Groningen
Argonne National Laboratory

In developing large, complex software systems, object-oriented programming techniques can provide many benefits. In addition to using an object-oriented language, developers should also employ other techniques such as layers to fully obtain these benefits. This article discusses several of these design details that were used in developing a military logistics system called Enhanced Logistics Intra-Theater Support Tool.

Planning for the transportation of large amounts of equipment, troops, and supplies presents a complex problem for military analysts. Software tools are critical in defining and analyzing these plans. Argonne National Laboratory developed the Enhanced Logistics Intra-Theater Support Tool (ELIST) to assist military planners in determining the logistical feasibility of an intra-theater course of action. This article focuses on the object-oriented design strategies used in developing the latest version of this system. Details of the specific military, logistical algorithms that were implemented can be found in other sources [1].

ELIST Model Requirements

The military logistics community has successfully used the previous version of ELIST (v.7) in planning analyses and training exercises for a number of years [2]. Ongoing use of this system has led to requests for more detail, more capabilities, and increased flexibility. Users wanted to model the transportation of military cargo at the individual vehicle level with a much more detailed simulation than in the existing ELIST system. Because of the size and complexity of the new logistics transportation model, performance was also a primary consideration. ELIST needed to be more reliable with a more robust data storage and handling system to address increased data requirements. Therefore, in developing this new version, Argonne National Laboratory took advantage of the opportunity to perform a total redesign of the program architecture.

Multiple languages were used to implement the previous version of ELIST. Initially, Prolog was used for most of the data and model representations and computations. C components and libraries were used for computations, user interface, and integration. Although ELIST was a very successful application, this multi-language approach proved difficult and time-consuming to port and maintain.

For the new ELIST, the Java language

was selected for many reasons. Java supports object-oriented features such as encapsulation, inheritance, abstraction, and polymorphism. Using Java would solve many portability concerns because of the availability of Java virtual machines on multiple platforms. The standard Java developer's kit provides built-in packages for user interface, database-connectivity, and distributed processing that address many maintenance concerns. Java's memory management and exception handling schemes address reliability concerns. Oracle was chosen as the database management system for the new version of ELIST because it would address many data storage requirements and was already in use at sponsor's sites.

Object-Oriented Design Approach

We chose evolutionary delivery for our life-cycle model [3]. Under this approach, we developed the new version of ELIST, showed it to users, and refined the software based on their feedback. The first step was to specify all of the logistical algorithms in a requirements document based on knowledge gained during prior model development and from interaction with the user community.

Based on these algorithms, we created Unified Modeling Language (UML) diagrams of the basic simulation objects. Using these requirements, we put our initial emphasis on developing the visual aspects of the system needed to support the data required by the simulation. As full functionality was added to these areas, it became apparent that more than a thousand classes would be required in the complete system.

In structuring an application of this complexity, we needed to employ a scheme for partitioning the software into manageable sections. We chose to use class-type architecture for our design [4]. In class-type architecture, the classes of the application are organized into well-defined layers based on their general function. Figure 1 shows the overall architecture of the ELIST system.

Each layer is well modularized and addresses a specific area of responsibility. The different layers can be developed relatively independently with an interface specifying their use by other layers. In designing each of these layers, we followed the recursive/parallel model [5], dividing each layer into subcomponents and gradually refining the classes as development progressed.

This design approach has many advantages. Changes to one layer are isolated from other layers, making the application more portable, extensible, and maintainable. In addition, different software teams can concentrate on different layers, drawing on their areas of expertise. Many of these independent layers can be structured as general-purpose packages in a code repository used across multiple projects. This approach enabled us to leverage the development efforts across multiple projects, saving expense and increasing code reliability.

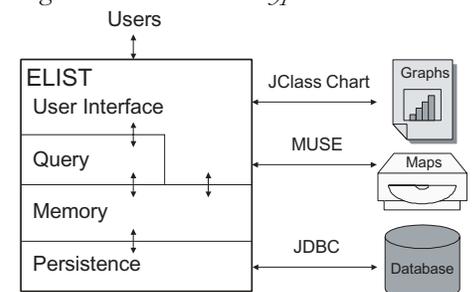
ELIST is composed of four main layers: the user interface layer, query layer, memory layer, and persistence layer. In each layer, UML was used to define classes and the relationships among them. Each layer presented its own set of issues that needed to be addressed in organizing the classes. In the sections that follow, each layer is discussed in detail, focusing on some of the techniques used in that layer.

ELIST is composed of four main layers: the user interface layer, query layer, memory layer, and persistence layer. In each layer, UML was used to define classes and the relationships among them. Each layer presented its own set of issues that needed to be addressed in organizing the classes. In the sections that follow, each layer is discussed in detail, focusing on some of the techniques used in that layer.

User Interface Layer

The topmost layer of the ELIST application is the user interface layer. Written using the Java Foundation Classes, this

Figure 1: *ELIST Class-Type Architecture*



layer presents graphical windows to the user. In developing the window designs, we prototyped windows and presented them to the user community for iterative feedback before actual code development began.

ELIST requires both traditional widgets such as tables, as well as custom widgets such as specialized trees and Gantt charts. An extensive package of generic, user interface widgets was developed for several reasons. One is that the standard Java widgets contain a large number of bugs. By developing our own widgets that map to these standard widgets, we were able to provide the bug fixes that were required as well as add custom features to the widgets. As new versions of Java are released, we will update only the user interface layer to accommodate any changes; this greatly increases maintainability of our models.

The commercial tool called JClass Chart from Sitraka was accessed to create standard graphs using a package within the user interface layer. Again, this allows us to switch tools if needed and add functionality beyond that supplied in the tools.

Most of our geographical information system (GIS) requirements could be implemented by writing a package that uses the 2D graphics package provided in the standard Java system. However, to display images created from standard map products, we wrote an interface on top of National Imagery and Mapping Agency (NIMA) Mapping, Charting, and Geodesy, Utility Software Environment (MUSE) software using the Java Native Interface utility.

The MUSE library provides routines for reading and writing standard NIMA map products. This gives us the flexibility to completely integrate our map windows

Figure 2: *Query Window*



with other parts of our application while taking advantage of existing code for reading the map products. In implementing this package, we used the technique called *wrapping*. Object-oriented classes were written to interface to non-object-oriented functions within a library.

A main editing window was available in the interface for each of the main objects in the memory layer. Each of these top-level windows organized the data for that object and provided multiple, related tabbed panels of information.

Query Layer

When dealing with huge amounts of data, users need a dynamic, flexible mechanism for retrieving subsets for various types of processing such as viewing, modifying, or tallying results. We developed the query layer to provide users with a way to build, save, retrieve, and execute complex queries about their data. When executed, each query returns collections of objects that match a defined premise.

The query package provides generic query and data assignment capability. Any object that is to be queried must publish what information can be retrieved or modified, and what data values are valid by implementing the *QueryObject* and *QueryObjectSummary* interfaces. The query system does not need to know any other information about the structure or function of the objects.

We designed the query package in three sub-layers: user interface, logical operations, and data management. The query package dynamically creates a window that allows users to construct queries and data assignments based on the information published by the data objects. Through user interface windows, users build arbitrarily complex expressions by nesting simple predicate expressions in a tree-like structure, as shown in Figure 2.

When the user is creating this tree in the interface, the system builds a corresponding hierarchy of *PredicateExpression* classes and *ConditionalStatement* classes in the logical operations layer. The first allows the construction of arbitrarily complex expressions while the second allows modification of data values within an object.

After the logical operations classes are created, the data support and management layer performs the query on the set of data objects. These data objects are typically in memory but may optionally be in a relational database. In this case, the query package can retrieve and store data in a relational database via Structured Query Language (SQL) statements using query keys that have been mapped to database fields. The

predicate expression generates the *where* clause of a SQL statement, which is then sent to the *PersistenceBroker*, which in turn builds the complete SQL statement and executes it.

Memory Layer

The heart of ELIST is its simulation, so in designing its memory layer (or business layer), the simulation's requirements were our primary concern. In examining the data requirements, we found that data can be divided into a number of main objects that have dependencies on other objects. Figure 3 shows the main objects in ELIST's memory layer.

These main objects represent logical divisions in the data. The user interface was structured to correspond to this division of objects by creating one main editing window for each of these objects. As shown by the arrows in Figure 3, each main object may depend on other objects. To support handling these dependencies, a Java interface *DependentObject* was defined. Each main object implemented this interface. By redefining basic methods in the interface, each object specified which other objects it depended on. This gave us a scheme for easily checking which objects were affected by changes in other objects. For example, if a user wants to edit a new network, we could quickly determine that any currently loaded scenarios would have to be unloaded. This enabled us to keep the object dependencies in the memory layer rather than hard-coding it in the user interface.

Metadata for each of these main objects were mapped to corresponding database tables that could be managed through tables in the user interface. Important metadata included descriptions, modification dates, owners, and classification levels. Including these data in our design enabled users to more easily track changes being made for different strategic plans.

Whenever objects are edited in the user interface windows, the corresponding objects are immediately changed in the memory layer, but not in the database. To support this feature, classes were developed that implement a *ChangeLog*. When a text field or other widget is edited, the corresponding memory layer objects are changed and a change record is created. All changes, whether updates, adds, or deletes, are stored in a queue associated with the window. When the user explicitly requests a save, this log is then used to propagate the updates to the database through the persistence layer. Special group records allow a set of changes to be grouped together. The user can display an undo log at any time and

may roll back changes in memory.

Figure 4 shows a UML diagram containing the main *ChangeLog* classes. As objects are edited in the user interface, methods in the *ChangeLog* class create instances of the appropriate type of *ChangeRecord* object.

Persistence Layer

Proper object-relational integration requires a strategy for mapping the object model to the relational model in order for Java objects to become persistent (saved for later use) in a relational database management system (RDBMS). Without some strategy, objects cannot be directly saved to and retrieved from relational databases. This problem of trying to maintain consistency between the objects in memory and the state of the database leads to writing hundreds of lines of embedded SQL code for reading and writing to the database.

There is a standard package available in Java for interfacing with commercial relational databases called Java Data Base Connectivity (JDBC). This package allows applications to connect to a wide variety of database products in a standard way. However, JDBC is still a lower-level application programming interface that does not facilitate a nice, modular encapsulation of the mapping needed to make memory layer objects persistent. To fully support our class-type architecture, we implemented a persistence layer that wraps the lower-level functionality of JDBC [6]. This provides a means for the objects in memory to create, retrieve, update, and delete themselves in the database. Figure 5 shows the main classes defined in the persistence layer.

Every object that needs to be persistent is a subclass of *PersistentObject*. The *ClassMap* class is defined to map an object to a table in the relational database. It separates the persistence mechanism from the object schema. In implementing these objects, a standard was adopted in which a subdirectory called *classmap* was defined under each package directory containing *PersistentObjects*. The corresponding *ClassMap* classes for those objects were stored in that subdirectory. For every type of *PersistentObject*, a *ClassMap* instance is created that stores the information needed to create *SELECT*, *INSERT*, *UPDATE*, and *DELETE* SQL statements and records information on the database table and columns used. The *ClassMap* object implements the database access for the corresponding *PersistentObject*.

The main class in the persistence layer is the *PersistenceBroker* class. This object acts as the database manager for ELIST, maintaining the connection to the RDBMS. It han-

dles communication between objects in the application and the persistence mechanism by wrapping the actual calls to JDBC. The *PersistenceBroker* holds the collection of *ClassMaps* for all *PersistentObjects* in memory. By using calls to JDBC, the *PersistenceBroker* class implements *saveObject*, *retrieveObject*, and *deleteObject* methods. It also implements a *processSQL* method that can submit any arbitrary SQL call.

When the user is editing data, the persistence layer works in conjunction with the *ChangeLog* mechanism. When a user selects a save option from an editing window, the *ChangeLog* for that window is used to forward those saves to the appropriate *PersistentObjects*. The *PersistenceBroker* finds the corresponding *ClassMap* for that class of *PersistentObject* and calls it to construct the appropriate SQL statement for the

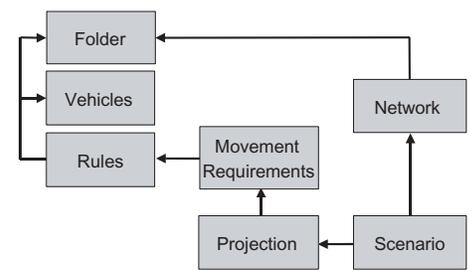


Figure 3: Dependencies of Main Objects

object. It then attempts to process the SQL statement using JDBC. If there are errors, the database is rolled back, a *PersistenceException* is thrown to the user interface layer; otherwise, the transaction was successful and the changes to the database are committed. Through the use of this exception handling mechanism, we were able to keep the persistence layer sep-

Figure 4: ChangeLog Classes

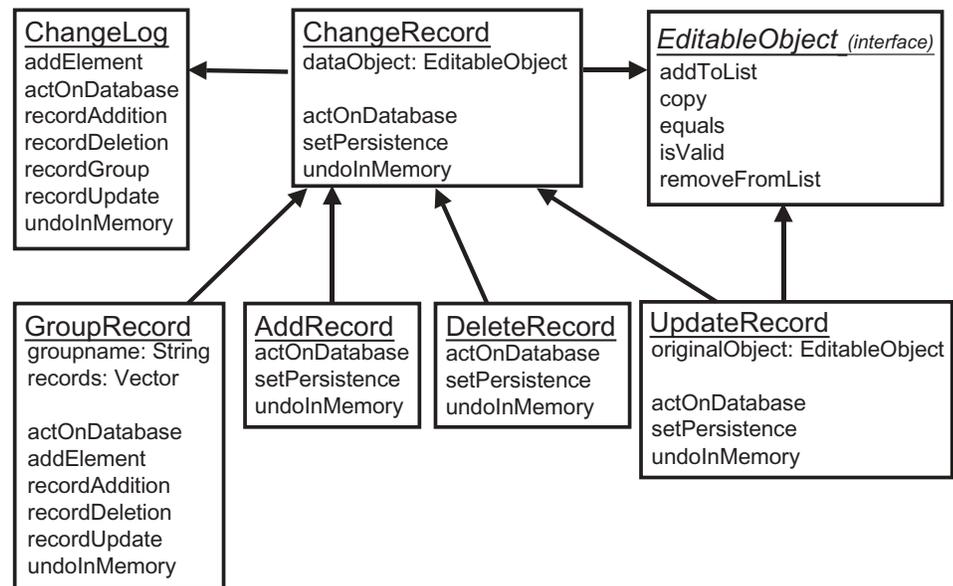
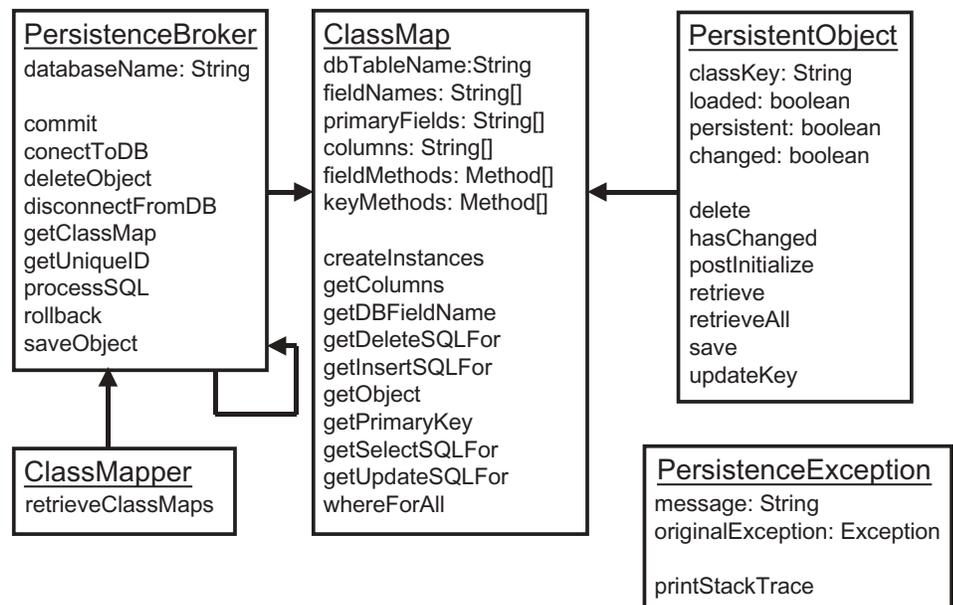


Figure 5: Persistence Layer Classes



arated from the user interface layer, and at the same time, keep the database in sync with the objects in memory.

Summary

Through ELIST development, we learned that it is essential to apply object-oriented techniques throughout many levels of our design. In addition to using an object-oriented language, we structured our application using class-type architecture. By dividing our application into layers, we were able to focus on separate, reusable components and assign lead developers to each layer who specialized in the respective component areas. By carefully designing each layer using UML modeling techniques, we addressed our primary concerns regarding portability, maintainability, and reusability. The resulting ELIST system has been successfully delivered to the sponsor and is evolving in response to new and refined requirements. The packages developed to support the various layers have been reused on multiple government projects, providing substantial cost savings for those development efforts as well. ♦

Acknowledgment

This work was supported under a military interdepartmental purchase request from the U.S. Department of Defense, Military Traffic Management Command Transportation Engineering Agency, through the U.S. Department of Energy contract W-31-109-ENG-109.

References

1. Braun, M.D., and C. Van Groningen. ELIST 8 Transportation Model. ANL/DIS/02-1. Argonne, IL: Argonne National Laboratory, 13 Feb. 2002.
2. Macal, C., C. Van Groningen, and M. Braun. Simulation of Transportation Movements Over Constrained Infrastructure Networks. Proc. of the 1995 Simulation Multi-Conference, Phoenix, AZ, 27 Apr. 1995 (4): 97-102.
3. McConnell, Steve. Rapid Development: Taming Wild Software Schedules. Redmond, WA: Microsoft Press, 1996.
4. Ambler, Scott W. Building Object Applications That Work: Your Step-by-Step Handbook for Developing Robust Systems with Object Technology. New York: Cambridge University Press, 1998.
5. Berard, Edward V. "Understanding the Recursive/Parallel Life-Cycle." Hotline of Object-Oriented Technology 1.7 (May) 1990: 10-13.
6. Ambler, Scott W. "Mapping Objects to Relational Databases." White Paper. AmbySoft Inc., 26 Feb. 1999.

About the Authors



Mary Ann Widing is an information systems engineer in the Decision and Information Sciences Division at Argonne National Laboratory. Her work at Argonne has focused on developing complex, graphical user interfaces for decision support systems used by government agencies. Widing has a Bachelor of Science and Master of Science in engineering from the University of Illinois in Urbana-Champaign.

Argonne National Laboratory
9700 South Cass Ave.
Argonne, IL 60439-4832
Phone: (630) 252-3798
Fax: (630) 252-6073
E-mail: widing@dis.anl.gov



Kathy Lee Simunich is a computer scientist in the Decision and Information Sciences Division at Argonne National Laboratory. Her work at Argonne includes environmental modeling and object-to-relational databases, as well as writing reusable components across various Department of Defense and Department of Energy projects. Simunich has a Bachelor of Science in meteorology from Northern Illinois University and a Master of Science in computer science from North Central College in Illinois.

Argonne National Laboratory
9700 South Cass Ave.
Argonne, IL 60439-4832
Phone: (630) 252-3285
Fax: (630) 252-6073
E-mail: simunich@dis.anl.gov



Dariusz Blachowicz is a computer scientist in the Decision and Information Science Division at Argonne National Laboratory. His work at Argonne includes a wide range of modeling and simulation applications, and Web-based interactive applications for Department of Defense and

Department of Energy agencies. Blachowicz has a Bachelor of Science in civil engineering and a Master of Science in computer science from Illinois Institute of Technology in Chicago.

Argonne National Laboratory
9700 South Cass Ave.
Argonne, IL 60439-4832
Phone: (630) 252-6187
Fax: (630) 252-6073
E-mail: blach@dis.anl.gov



Mary Braun is a computer systems engineer in the Decision and Information Sciences Division at Argonne National Laboratory. Her work has focused on military logistics modeling and simulation. Braun has a Bachelor of Science from the University of Santa Clara and a Master of Science from the University of California, Berkeley, both in electrical engineering.

Argonne National Laboratory
9700 South Cass Ave.
Argonne, IL 60439-4832
Phone: (630) 252-3727
Fax: (630) 252-6073
E-mail: duffy@dis.anl.gov



Charles Van Groningen, Ph.D., leads the Enhanced Logistics Intra-Theater Support Tool development team at Argonne National Laboratory. His research interests include modeling, simulation, and knowledge representation. Van Groningen has a doctorate in artificial intelligence from the Illinois Institute of Technology, a Master of Science in computer science from DePaul University, and a Bachelor of Arts in mathematics from Trinity Christian College.

Argonne National Laboratory
9700 South Cass Ave.
Argonne, IL 60439-4832
Phone: (630) 252-5308
Fax: (630) 252-6073
E-mail: vang@dis.anl.gov



Extreme Software Cost Estimating

Dr. Randall W. Jensen
Software Technology Support Center

One dominating software development complaint is the inability to estimate cost, resources, and schedule with acceptable accuracy. Several methods of schedule and cost estimation have been proposed during the last 25 years with mixed results due, in part, to the focus and capability limitations of traditional estimation models. A significant part of estimation failures can be attributed to not understanding the inner workings of the software development process and its impact on the parameters used in schedule and cost estimates. For example, poor management can increase software cost, schedule, and quality more rapidly than any other variable, while good project management can decrease development cost and schedule just as rapidly. This article describes a management-centric approach to predicting software development cost and schedule in a modern, or extreme, development environment as opposed to the traditional technology-based approaches. Techniques necessary to produce realistic, reliable software development estimates are introduced, as well as quantitative methods for predicting the impacts of management decisions. This article was awarded the Outstanding Software Paper at the International Society of Parametric Analysts 2003 Annual Conference in Orlando, Fla.

The software estimating tools in widespread use today evolved from models developed in the late 1970s to early 1980s using project data available at the time. These widely used tools include the Constructive Cost Model (COCOMO) II, Price-S, Sage and SEER-SEM. It is important to note these mature tools are as useful today as they were 20 years ago when they were first formulated. Input data parameter sets (analyst and programmer capability, application experience, use of modern practices and tools, etc.) developed for Seer to describe organizations in the early 1980s are, oddly enough, still generally applicable today. Fortunately for the estimating model developer, culture changes very slowly, if at all.

We have been able to learn new things about software development during this period. For example, Barry Boehm wrote the following in 1981:

Poor management can increase software costs more rapidly than any other factor. Each of the following mismanagement actions has often been responsible for doubling software development costs ... [1]

Of course, you have to read the first 485 pages of his book to get to this logical, yet profound statement. Most readers do not seem to get that far. Gerald Weinberg's Second Law of Consulting [2] added a supporting observation: "No matter how it looks at first, it's always a people problem."

There have been several development technology breakthroughs during the past 40 years that have significantly decreased the cost of software products. For example, the introduction of FORTRAN and

COBOL decreased the cost of a given product functionality to one-third of the cost when implemented in Assembler. The transition from C++ to the newer visual languages, and the advent of object-oriented structures created additional large savings in product cost.

However, when we look at the effort required to produce a single line of source code in any given programming language, we see that software development productivity (measured from start of development through software-system integration) has increased, with little blips and dips, almost linearly at the rate of less than one source line per person-month per year as shown in Figure 1. The aged heuristic, which portions the development effort into design, code, and test (40-20-40 = 100%), shows that eliminating the coding activity entirely leaves 80 percent of the work remaining. The advent of powerful programming environments primarily affects only the coding activity.

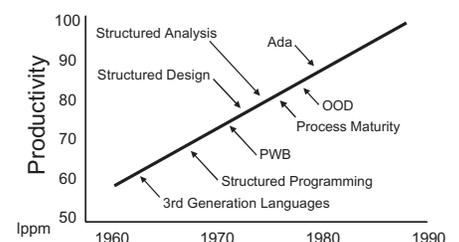
The importance of people shows up in the literature as early as the Hawthorne study by Elton Mayo [3]. This work showed people are primarily driven by esteem and self-actualization, and not by physiological and safety needs (*Rabble hypothesis*). The work of Mayo paved the way for the development of the classic *Theory X – Theory Y* proposal by Douglas McGregor [4] and the Herzberg motivators [5]. W. E. Deming [6] extended these ideas with his *total quality management* work in Japanese and American industry. In spite of the work by these behavioral pioneers and many others, software management remains what Herzberg refers to as a *Theory X* culture. Scott Adams' Dilbert cartoon character and DeMarco's "Covert

Agenda"¹ are two examples of the existence and dominance of this culture.

There are three important dimensions in software management: project, process, and people, as shown in Figure 2 (see page 28). Project was the primary software development focus in the 1960s when the software development discipline was new. The early 1970s brought a shift in focus to the development process. The emphasis on the Waterfall Model in software development, defined and enforced through standards such as Mil-Std-2167A, began a trend that is still flourishing today. The mid-1980s introduced the Software Engineering Institute's Capability Maturity Model[®] as an approach to stabilizing the development process and improving quality and productivity. By focusing energy on process improvement, we can ignore the importance of people in the development process. "Get the process right and people are interchangeable" is a common battle cry. Process is a necessary element of process improvement, but not sufficient to solve the software productivity problem.

Recent developments in teaming concepts led to a focus on management and people issues. The introduction of extreme and agile development methods demonstrated the importance of management

Figure 1: *Software Development Productivity Gains: 1960 to 1990*



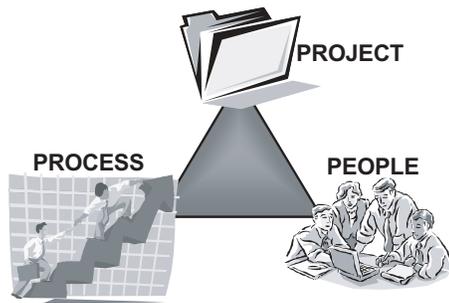


Figure 2: *Project-Process-People Triangle*

and people issues in development productivity and quality. Unless people are considered an important part of the project-process-people triad, software development cost and schedule estimates will continue to be inconsistent and unstable.

Agile Software Development

The Manifesto for Agile Software Development [7], first published Feb. 13, 2001, states the following:

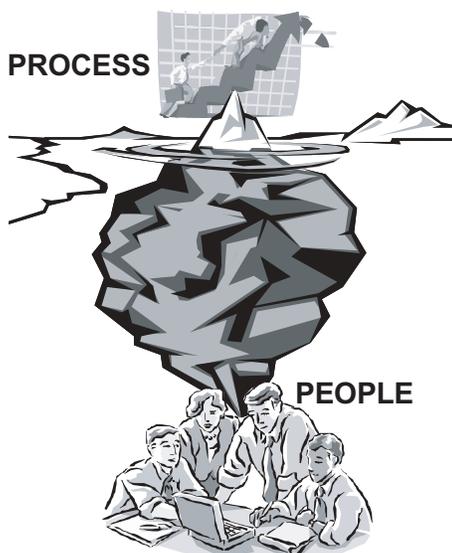
We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions over processes and tools.
- Working software over comprehensive documentation.
- Customer collaboration over contract negotiation.
- Responding to change over following a plan.

That is, while we value the items on the right, we value the items on the left more.

The bulk of the work in the design and test activities (again, 80 percent of the total)

Figure 3: *Process Is But the Tip of the Project Iceberg*



involves a high level of communication that is impacted by the environment, the people, and the organization as well as the development process.

Tonies' Effectiveness Formula

Chuck Tonies introduced the concept that the effectiveness of a software engineer is more than IQ, training, and experience in the 1979 text "Software Engineering" [8]. He pointed out that people in software-related positions in industry work in highly interactive environments. The software development *team* consists of programmers, analysts, test engineers, managers, customers, and users to name a few of the participants.

I italicized team to emphasize the two levels of teams: a group of people assigned to a project (the normal use of the term), and a team in the sense of a professional basketball team. The team in italics suggests the first level: people working as a unit even though their teamness is simply a common charge number and a loose relationship among the players in the project. The second team-level type involves a tight, highly communicative relationship, which is difficult to perceive when all of the members are isolated in cubicles like Dilbert and his cartoon co-worker, Wally.

Members of the development team may be cast in one or more of the roles involved in a project. It is important that people are aware of activities around them and understand their relationship to these activities to achieve their highest effectiveness. They must understand and act in concert with the project management plan, which includes communicating coherently with the other people assigned to the project. However, if the team (either definition) members are unwilling or not motivated to participate in sending or receiving information about the task at hand, the members' technical contribution to the project will be diminished, no matter how gifted or brilliant the individuals are.

Some degree of change is present in almost every development, even the stable projects. The complexity of software development carries with it incomplete and incorrect interpretations of the requirements, interface, and designs. Constant communication among the participants is the only way misunderstanding and errors can be corrected. The danger of emphasizing the process over people and communications is a major point in the Agile Manifesto as shown in Figure 3. Process is only the *tip of the iceberg*, with people making up the bulk of the iceberg.

Tonies postulated that an individual's

value to the development organization in an industrial environment depends on three attributes: computer science skills, communication skills, and management skills. The product gives the effectiveness of the individual in the organization in the following equation:

$$E = CS \times C \times M \quad (1)$$

where:

E = net effectiveness

CS = computer science technical skills (0-1)

C = communication skills (0-1)

M = management skills (0-1)

The effectiveness formula in Equation (1) shows that if any of the three elements are missing, the effectiveness approaches zero. Our experience in the software product-centered environments shows it a realistic model of software engineering performance. It is true that we live in an age of technical specialization. It is also true that software development and engineering is by its nature a complex interactive process that requires careful intensive management. The manager must contribute to the free exchange of information among software development players.

Boehm's list of management problems describes the common software management style for that time – a style that is prevalent today. He states the following:

Poor management can increase software costs more rapidly than any other factor. Each of the following mismanagement actions has often been responsible for doubling software development costs ... Despite this cost variation, COCOMO does not include a factor for management quality, but instead provides estimates that assume the project will be well managed. [9]

G. M. Weinberg [10] extended this discussion by grouping cost impacts described in Boehm's "Software Engineering Economics" to illustrate the relative importance of each impact group. Figure 4 presents Weinberg's results emphasizing the relative importance of organization and management in projecting software development cost. The people impact in Figure 4 represents education, IQ, and experience. Weinberg also points out in this text reference an interesting relationship between the Software Engineering Institute's research publications and relative cost impacts.

The people facet in Figure 2 includes

the most important of the three management facets in terms of productivity and quality gains, and represents the bulk of the communications and management factors that Tonies states.

Traditional Estimating Methods

Traditional estimating methods focus on the technical aspects of software development: project and process. An example of the traditional focus is the intrinsic capabilities of the analysis and programming team members. The principle measures of analyst quality are ability (education, intelligence, and problem solving skills), efficiency and thoroughness, and team communication.

The capability definition deals with capabilities in terms of the *team*; the interpretation generally is a collection of individuals working on a development activity. We abstractly discuss the concept of a team, yet when we look at the project environment, we see a *cube farm*² or a group of people working in isolated offices or widely dispersed locations.

Notice the traditional definition of capability lists cooperation and communication as one of three primary measures, but never mentions the factors that produce esteem and self-actualization; that is, motivation and management.

Extreme Software Estimating Methods

Traditional estimating methods are largely based on Theory X management methods. That is, the soft, or organization aspects of the environment, are difficult to measure and are to be avoided. Boehm said as much in “Software Engineering Economics” [11].

I also avoided the soft factors in the Seer years because of their assessment difficulty. However, I found many projects over a 20-year period that defied reconciling actual cost and schedule results with estimates. It was often impossible to turn the knobs on the estimating models to obtain a cost or schedule match. Once enough data was available to conduct an analysis, I found that all of the abnormally successful projects (higher productivity, etc.) had a common thread – Theory Y managers managed the projects. The problem remaining was to find a way to evaluate organization management. The measures are rather obvious (when outside the box) and easy to measure.

Several factors can be used to assess the capability of an organization: (1) motivation and management style, (2) use

of team methods and proximity of team members, and (3) information flow in the development environment. The remaining traditional capability factors are problem solving ability and programming skills.

Motivation and Management Style

Motivation is one of the most effective and important tasks facing any manager as shown in Figure 5. This task becomes critical in managing a creative, communication-centered activity such as software development. Management style must be considered before other improvement areas since it is the basis for both team concepts and working environment.

Theory X managers manage by control (as directors), closely supervise their employees, and are devoted to structure in both organization and process. Those who search for tools and methods to solve the productivity and quality problems are inherently traditional Theory X personalities. Theory X also underlies the concept that people are interchangeable if the development process is defined and stable.

Human behavior according to Theory Y is quite unlike Theory X behavior. Properly motivated people can achieve their own goals best by directing their efforts toward organizational goals. Theory Y people are motivated at the self-actualization, social, and esteem levels rather than the physiological and safety levels as assumed in Theory X. If the workers have little process ownership, the process is unlikely to change.

The importance of motivation in the development organization is much greater than the space devoted to it here. It is a topic worth additional study by those searching for major gains in quality and productivity.

Team Methods and Proximity of Team Members

A good example of a team approach that did not work is the Chief-Programmer Team [12] introduced by IBM in the 1970s. The team consisted of a chief programmer (creative, good problem solver, intelligent, etc.), a backup programmer (backup and insurance in case the chief programmer became incapacitated or went to the competitor for higher wages), functional specialists (dealt with narrow issues outside the chief programmer’s expertise), coders to implement the architecture and design, and a librarian to keep track of all the stuff being developed.

The team structure was *controlled central-*

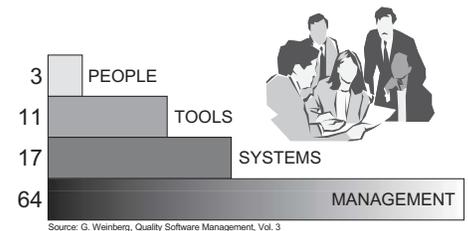


Figure 4: Relative Impacts of Development Environment Elements on Software Costs

ized, meaning top-level problem solving and team coordination are the responsibility of a team leader. Communication tended to be vertical. The chief programmer planned, coordinated, and reviewed all technical activities. This team structure had all the elements necessary to satisfy a high capability rating for the organization.

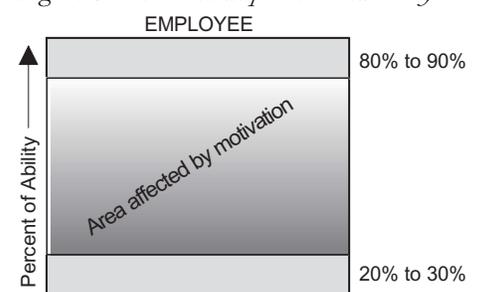
However, the concept failed. Why? First, the team was sensitive to the nature of the chief programmer, which helped to create a low morale environment. Second, the chief programmer team failed to congeal into a team (second definition). It is interesting to note the chief programmer team was still listed as one of the 10 most important ideas in software engineering in 1972 and 1982 by Construx Software Builders, Inc. in a 2002 keynote address³.

Information Flow in the Development Environment

A project’s productivity is tightly related to “how long it takes for information [to flow] from one person’s mind to another’s [13].” There are a number of factors to consider when evaluating information flow (or *convection* as Cockburn describes flow). The obvious measures are distance between developers and noise, including background noise; that is, sound not related to the task at hand. The best representation of good information flow is two people working at a whiteboard. This communication channel contains the best of good communication features: visual cues, visual persistence, sensation of movement, sound, timing (real-time questions and answers), and emotion.

Locating multiple projects (tasks) within the project area creates significant noise.

Figure 5: Motivation Impact on Productivity



Interruptions are also significant flow problems. Other less obvious information flow disrupters are doors and aisles. Telephones and e-mail are useful approaches to decrease information flow. Some programmers are information radiators, such as a programming language consultant located in a project area. Information radiation can take place as information displayed where the developers can readily see it. Walls are common locations for radiators. Note: Web pages are not information radiators.

Other programmers tend to be information sinks. Sinks include people who do not participate in the circulation of information. The infamous lone programmer who works alone, behind a closed door or a closed mind is a typical sink that demonstrates restricted information flow.

Estimating Method Needs

Projects can only be described through input parameters. Estimating tools cannot conjure information that has not been supplied by the estimator. The question we must ask ourselves is, can my estimating tool account for the following:

- Organization and management style (Theory X/Y).
- Motivation.
- Team use.
- Development environment (cube farms, skunk works).

Summary and Conclusions

Traditional estimating methods have worked well in the past because 90 percent of software projects have been developed by traditional organizations. Boehm's assumption that management style and capability could be ignored was generally true in 1981 when COCOMO was initially released. The term *well managed* was an overstatement, and still is, for most development organizations. Consistent management has become a better process descriptor than well managed in the focus era.

Well-managed projects, using the Tonies effectiveness formula, are still the exception rather than the rule. Traditional estimating methods and tools will continue to work in the near future because the style change is risky and very difficult.

Traditional estimating methods also benefit from organization stability. No change in organization style equates to no need for change in estimating approach or tools. Traditional estimating tools use fewer estimating parameters because management and communication effects can be ignored. Last, but not least, there is one estimate area that can be avoided – evaluation of the organization's management

style and effectiveness.

Extreme software estimating methods are needed because accurate software development estimates require more robust estimating models. Ignoring management style and motivation produces high schedule and cost estimates in modern organizations, and produces low estimates in poorly managed organizations. The most important estimating parameter is ignored, or poorly treated, in traditional approaches.

Competitive pressures are forcing organizations to rethink their approaches to effective software development. The number of software projects developed by modern, and possibly agile, organizations is rapidly increasing and driving a need for more estimating flexibility. Extreme estimating methods and tools provide a level of visibility in organization effectiveness that encourages both process and organization improvement. ♦

References

1. Boehm, B. W. Software Engineering Economics. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1981: 486.
2. Weinberg, G. W. The Secrets of Consulting. New York: Dorset House Publishing, 1985: 5.
3. Mayo, Elton. The Human Problems of an Industrial Civilization. New York: The MacMillan Company, 1933.
4. McGregor, Douglas. The Human Side of Enterprise. New York: McGraw-Hill Book Company, 1960.
5. Herzberg, Frederick. Work and the Nature of Man. New York: World Publishing Co., 1966.
6. Deming, W. Edwards. Out of the Crisis. Cambridge, Mass: MIT Press, 1982.
7. Fowler, M., and J. Highsmith. "The Agile Manifesto." Software Development Aug. 2001.
8. Jensen, R. W., and C. C. Tonies. Software Engineering. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1979: 24ff.
9. Boehm 486-7.
10. Weinberg, G. W. Quality Software Management: Congruent Action, Vol. 3. New York: Dorset House Publishing, 1994: 14.
11. Boehm 487.
12. Baker, F. Terry. "Structured Programming in a Production Programming Environment." IEEE Transactions on Software Engineering SE-1.2 (1975): 241-252.
13. Cockburn, A. Agile Software Development. Boston, MA: Pearson Education, Inc., 2002: 77.

Notes

1. *Covert Agenda*: To apply pressure to developers to get them to work longer and harder by the following:
 - Promote an ethic of workaholism.
 - Get project members to sacrifice personal lives.
 - Gull members into accepting hopeless schedules.
 - Hold members' feet to the fire to make them deliver.
2. A *cube farm* is a descriptive term for a facility in which the floor is divided into a large group of cubicles. Another term for this facility organization is a maze. Dilbert works in a cube farm.
3. "The 10 Most Important Ideas in Software Engineering." Construx Software Builders, Inc., 2002 <www.construx.com/docs/open/10MostImportantIdeas-Keynote.pdf>.

About the Author



Randall W. Jensen, Ph.D., is a consultant for the Software Technology Support Center, Hill AFB, with more than 40 years of practical experience as a computer professional in hardware and software development. He developed the model that underlies the Sage and the GAI SEER-SEM software cost and schedule estimating systems. He retired as chief scientist in the Software Engineering Division of Hughes Aircraft Company's Ground Systems Group. Jensen founded Software Engineering, Inc., a software management consulting firm in 1980. Jensen received the International Society of Parametric Analysts Freiman Award for Outstanding Contributions to Parametric Estimating in 1984. He has published several computer-related texts, including "Software Engineering," and numerous software and hardware analysis papers. He has a Bachelor of Science, a Master of Science, and a doctorate all in electrical engineering from Utah State University.

Software Technology

Support Center

6022 Fir Ave., Bldg. 1238

Hill AFB, UT 84056-5820

Phone: (801) 775-5742

Fax: (801) 777-8069

E-mail: randall.jensen@hill.af.mil



Software Insecta Zodiac

New Year is a time to ruminate on the past and gaze to the future, hoping to find destiny in the bygone. Two popular examples of this activity are the Chinese and Astrological Zodiacs. The only connection I found between software, China, and the stars was loads of late night Chinese take out prior to final installation.

However, after extensive research, I discovered a novel correlation between the character of software professionals, the month they developed their first application, and insects. I call this correlation the Software Insecta Zodiac in which insects represent each month of the year and impart distinct characteristics to those who initiated their software career in that month.

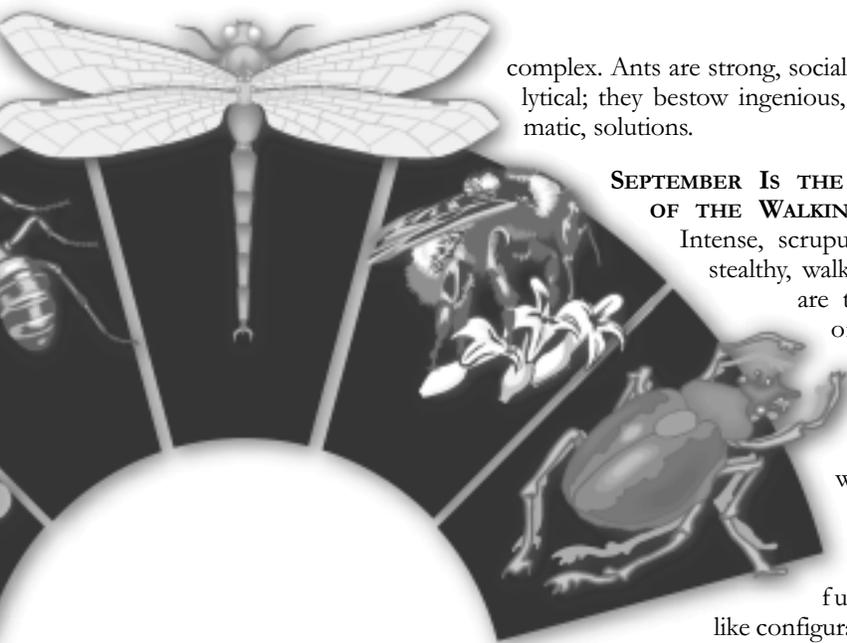
To tune in your karma, locate the month you developed your first software application and find enlightenment.

JANUARY IS THE MONTH OF THE HONEYBEE. Organized, diligent and industrious, honeybees are software manufacturers. They transform desire into reality. Honeybees meld architecture, design, language, platform, and constraints into sweet pliable solutions. They are true programmers.

FEBRUARY IS THE MONTH OF THE MANTIS. Powerful, deceptive, and predatory, mantes are software mercenaries. A mantis will design, code, or crack any system for a price. Thriving on big challenges mantes habitually prey on budgets.

MARCH IS THE MONTH OF THE LEPIDOPTERA. Flamboyant, chivalrous, and articulate, these moths and butterflies are highflying sages. Butterflies are fragile flighty egomaniacs that soar from job to job without conclusion. Moths are nimble, reliable, nocturnal gurus that stick with projects to closure.

APRIL IS THE MONTH OF THE LOCUST.



Communal, migratory, and voracious, locusts are project popinjays. Jumping from cubical to cubical, crickets deal gossip and grasshoppers party. In small numbers, they can gel a team. In swarms, they devour projects.

MAY IS THE MONTH OF THE MAYFLY. Intense, ostentatious, and impatient, mayflies are in the wrong profession. They seek riches in software only to find frustration. Mayflies that stay in the software business transform into mosquitoes that siphon life out of warm-blooded projects, organizations, and management.

JUNE IS THE MONTH OF THE BEETLE. Tough, persistent, and assiduous, beetles are the blue-collar staff of software. Rhinoceros beetles are headstrong, forceful engineers that push projects through tough challenges. Ladybugs are diligent reliable and perfect for security against intruding pests. Dung beetles scour project excrement to expose and tackle crucial obstacles.

JULY IS THE MONTH OF THE GADFLY. Persistent, caustic and irksome, gadflies are project critics. These backbiting gnats, piercing aphids, and parasitic fleas are curmudgeons that disparage decisions, derail design, and detour plans. Gadflies induce project paralysis.

AUGUST IS THE MONTH OF THE ANT. Strong, organized, cerebral, and disciplined, ants are software architects. They transform dreams into desire. They simplify the

complex. Ants are strong, social, and analytical; they bestow ingenious, yet pragmatic, solutions.

SEPTEMBER IS THE MONTH OF THE WALKING STICK.

Intense, scrupulous, and stealthy, walking sticks are the patrol officers of software. Seldom noticed, walking sticks perform critical functions like configuration control, inspection, and testing. Sticks keep projects in line and on target.

OCTOBER IS THE MONTH OF THE COCKROACH. Lascivious, sneaky and wasteful, cockroaches are software swindlers. Often holding multiple jobs, roaches pilfer and profligate hours, software, equipment, supplies, and eventually customers.

NOVEMBER IS THE MONTH OF THE TERMITE. Eccentric, intense, and resourceful, termites are software innovators. Termites destroy more than they build; yet in the end, their ingenuity rouses industry standards for the future.

DECEMBER IS THE MONTH OF THE DRAGONFLY. Energetic, aggressive, and inspiring dragonflies are software leaders. Dragonflies are pacesetters that seldom tire, offer expansive vision and prune the superfluous. Dragonflies inspire and deliver.

Do you work with any of these creatures? Pragmatist or mystic, your ability to recognize project detractors and contributors is crucial. Exterminate mayflies, gadflies, and roaches. Harness the flair of mantes, locust, and termites. Yield direction to moths and dragonflies, support walking sticks, and employ more ants, beetles and honeybees. In short, get your entomologic entourage together.

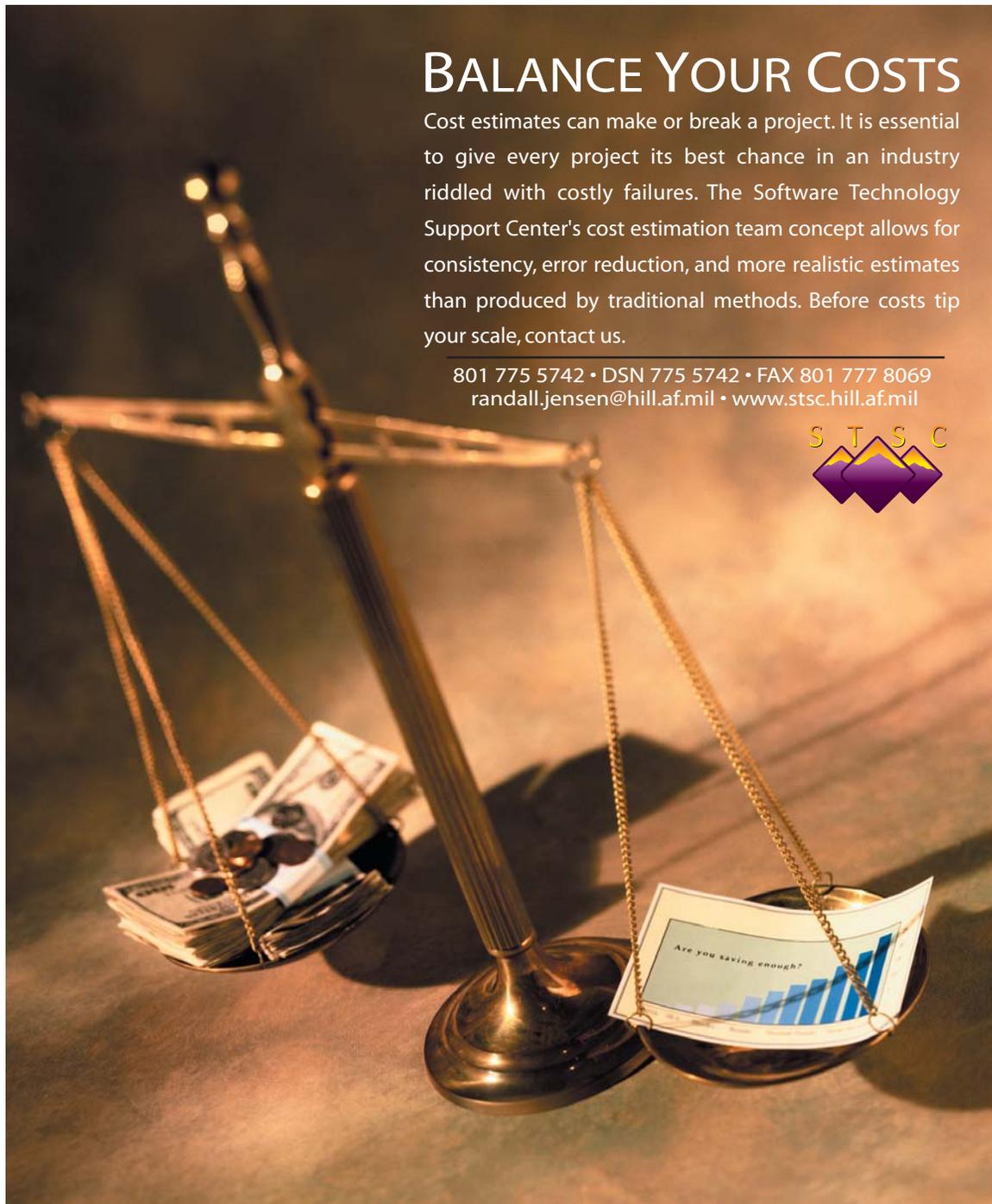
– Gary Petersen
Shim Enterprise, Inc.



BALANCE YOUR COSTS

Cost estimates can make or break a project. It is essential to give every project its best chance in an industry riddled with costly failures. The Software Technology Support Center's cost estimation team concept allows for consistency, error reduction, and more realistic estimates than produced by traditional methods. Before costs tip your scale, contact us.

801 775 5742 • DSN 775 5742 • FAX 801 777 8069
randall.jensen@hill.af.mil • www.stsc.hill.af.mil



Published by the
Software Technology
Support Center (STSC)

CROSSTALK / MASE
6022 Fir Ave.
Bldg. 1238
Hill AFB, UT 84056-5820

PRSR STD
U.S. POSTAGE PAID
Albuquerque, NM
Permit 737