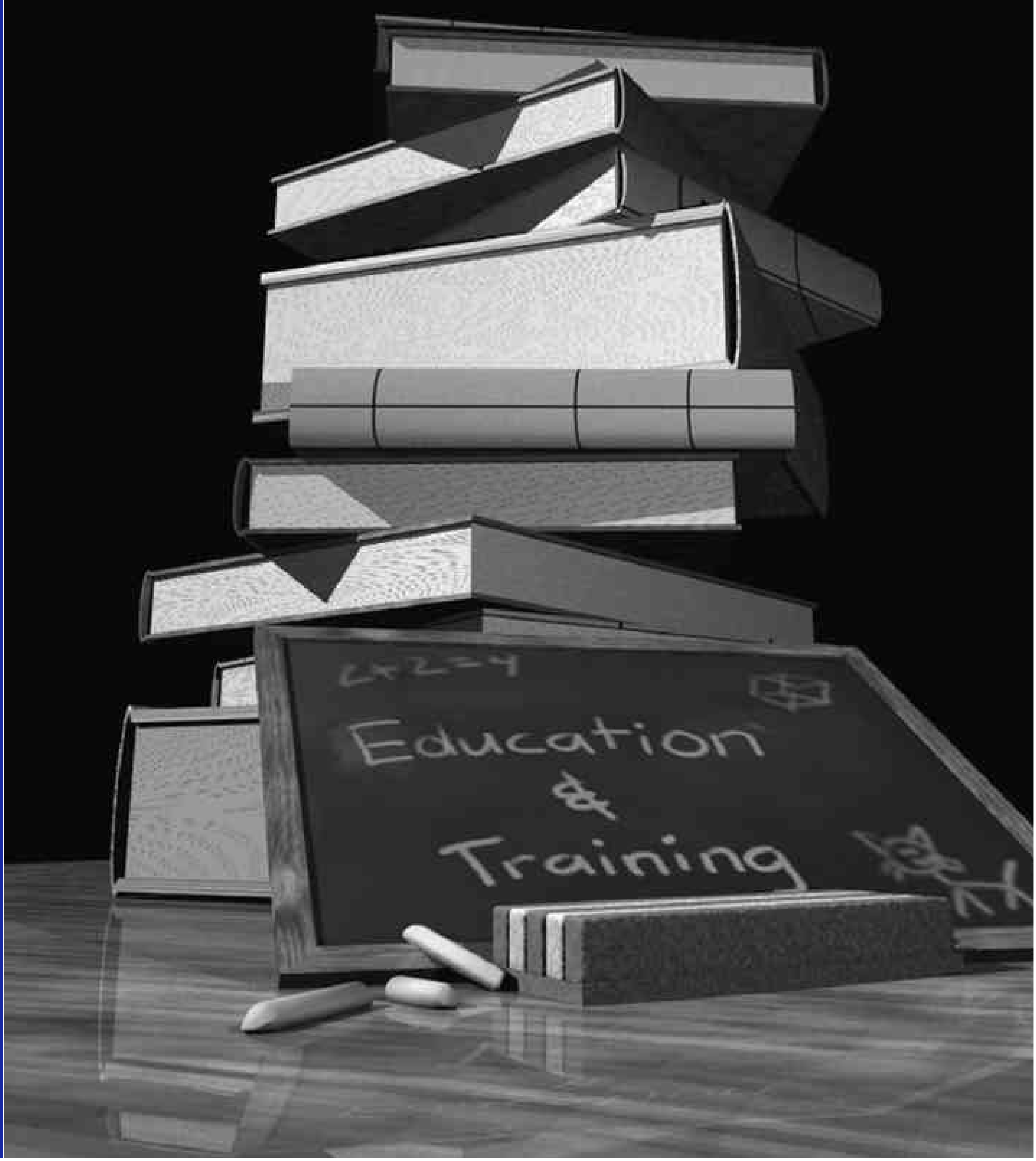
CrossTalk

March 2000 The Journal of Defense Software Engineering Vol. 13 No. 3



On the cover: Anthony Peters of L-3 Communications in Layton, Utah, represents education and training with a stack of books and old-fashioned chalkboard.

Departments

CrossTalk

Sponsor	Lt. Col. Joe Jarzombek
Publisher	Reuel S. Alder
Associate Publisher	Lynn Silver
Managing Editor	Kathy Gurchiek
Associate Editor/Layout	Matthew Welker
Associate Editor/Features	Heather Winward
Voice	801-775-5555
Fax	801-777-8069
E-mail	crosstalk.staff@hill.af.mil
STSC Online	http://www.stsc.hill.af.mil
CrossTalk Online	http://www.stsc.hill.af.mil/Crosstalk/crostalk.html
CRSIP Online	http://www.crsip.hill.af.mil

Subscriptions: Send correspondence concerning subscriptions and changes of address to the following address:
Ogden ALC/TISE
7278 Fourth Street
Hill AFB, Utah 84056-5205

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

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

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

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

STSC Online Services: at http://www.stsc.hill.af.mil. Call 801-777-7026, e-mail randy.schreifels@hill.af.mil.

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

The Software Technology Support Center was established at Ogden Air Logistics Center (AFMC) by Headquarters U.S. Air Force to help Air Force software organizations identify, evaluate, and adopt technologies to improve the quality of their software products, efficiency in producing them, and their ability to accurately predict the cost and schedule of their delivery.

Delivering Just-in-Time Training

Deputy Defense Secretary John Hamre has acknowledged that the Department of Defense (DoD) does not have a cutting-edge education system for acquisition officials. [1] Moreover, because software is so pervasive within the DoD systems, we need to dispel the myth that only software practitioners should attend Software Acquisition Management courses. An alternative would focus changes in system acquisition courses to include software-related topics. For acquisition certifications, anyone dealing with program management, systems engineering, or test and evaluation should be required to take courses that offer insight into the acquisition and support of software-intensive systems. Given the rate of change in practices and policy, this prompts the question of how DoD can deliver requisite, up-to-date education and training in a timely, relevant manner to our dispersed acquisition workforce.

Like other organizations that are challenged to manage rapid change, complexity and diverse, dispersed workforces, the DoD must continue to search for methods to increase efficiency and effectiveness, to do more with less. To achieve the involvement of all their people resources, many organizations are looking to virtual teams, the "learning organization," and the collaborative enterprise. Increasingly, communities of practice (human networks) are supported by electronic communication. Information technology offers power and reach, and an all-digital format offers more flexibility. Use of the Internet, by itself, has bandwidth limitations. Individualized, self-paced training tends to be media-rich. Interactive, media-rich training is best delivered via a CD-ROM, with access to enterprise networks and the Internet. Such a solution offers tremendous potential return on investment, leaving the central challenge in capturing and delivering information quickly and economically to the practitioners who require it when the material is relevant to their needs.

Just-in-Time CD-ROM-based training, with links to an intranet or the Internet, offers an efficient way to deliver education, training, or information to dispersed locations and is a key to managing rapid change in technology, business processes, policies, and strategies. The technology allows flexible, modular, and indexed presentations. Lectures and information can be constructed from many sources without undue advance preparation demands. Its modularity allows efficient search and retrieval of information. On a larger scale, this technology may be deployed on servers with master indexes and version control, allowing searches of vast bodies of information. An added value is that it facilitates a more flexible delivery to allow trainees or learners to select their own design of instruction and information retrieval that fits their learning objectives within the context of their needs and organizational environment.

Why is Just-in-Time CD-ROM-based training significant to the successful dissemination and use of any complex subject that requires periodic reference? Traditional training and information delivery methods cannot reach the vast majority of practitioners in a manner that provides the information when it is relevant to their needs.

Organizations can manage the environment necessary for interdisciplinary communities of practice to flourish and share information that is a product of that knowledge. For organizations to more successfully compete in an era of rapid change, they need to use Just-in-Time CD-ROM-based training and information delivery, with links to an intranet or the Internet. Using the leveraging capabilities of interactive multimedia, organizations can complement their own virtual knowledge management infrastructure that evolves as part of employees' routine information flow and supports interdisciplinary efforts. Just-in-Time CD-ROM-based training supports the growth of knowledge management, which lets practitioners learn what they need to know on the spot when they need it, and it supports the growth of informal learning that occurs outside of a formal context such as a classroom or a scheduled course.

This is my last publisher's note in my role as CRSIP Director, as I am retiring April 1 (see p. 6 for more on Lt. Col. Jarzombek). It has been a true learning opportunity to work with the professionals in the Software Technology Support Center, Air Force Research Laboratory, Software Engineering Institute, Software Engineering Process Groups, Office of the Secretary of Defense-sponsored teams, and the various conferences for which I have been involved. These groups have been instrumental in extending the state-of-the-practice and strengthening our software community of practice. I know that *CROSSTALK* will continue to be a vital conduit for information exchange within our software community.

Lt. Col. Joe Jarzombek, CRSIP Director

Reference

1. Hamre Warns Of Struggles Ahead In Support Systems, Security, Outsourcing. *Defense Information and Electronics Report,* May 14, 1999, p. 1.

Anywhere, Anytime Learning in DoD

Long a world-class trainer, the Department of Defense (DoD) is leading the way into the learning environment of the future.

Recognizing that today's communications, information, and learning technologies allow a unique opportunity to develop common solutions to common learning challenges, the DoD's approach has been to form close partnerships, not only across the military departments and defense agencies, but with other federal agencies, academia, and private industry. The Advanced Distributed Learning (ADL) Initiative is the DoD's vehicle for developing and applying learning technologies on a broad scale.

Last year, Deputy Secretary of Defense John Hamre directed the Under Secretary for Personnel and Readiness to develop an Advanced Distributed Learning Initiative to "identify more efficient and effective ways to educate, train and support DoD personnel." Secretary of Defense William S. Cohen published DoD's Training Technology Vision to "ensure that DoD personnel have access to the highest quality education and training that can be tailored to their needs and delivered cost effectively, anytime and anywhere." The White House issued an Executive Order that created a Federal Task Force to "recommend . . . policy to make effective use of technology to improve training opportunities for federal government employees," that cited DoD's ADL Initiative as a model for other federal agencies to follow. At Congress' request, the department published the *DoD Strategic Plan for Advanced Distributed Learning.*

ADL is not limited to education, training, or performance aiding. It encompasses them all. The ADL Initiative is designed to leverage the full power of communications, information, and learning technologies—through the use of collaboratively developed common standards—in order to achieve Secretary Cohen's vision. The engine of this exciting revolution in learning is the Internet, as it is rapidly becoming the engine of all contemporary commerce, communication, and learning. Furthermore, the ADL Initiative seeks to satisfy the needs of everyone associated with learning and performance, including teachers and students, administrators, developers, managers, parents, evaluators, and others. It involves a number of challenges.

The challenge to teachers is to understand and apply the new technologies in concert with, in addition to, or in some cases in place of, traditional learning methods. The challenge to developers is to design methods of instruction and content that are open-architecture, shareable, high quality, and cost effective. The challenge to the information technology sector is to field an infrastructure that supports anytime, anywhere learning with appropriate bandwidth, transaction security, and robustness that is transparent to the learner.

These are not easy challenges. They require unprecedented collaboration and consensus building. ADL, like so much of contemporary life, is about converging historically separate domains in order to reach a whole greater than the sum of its parts. Since the birth of distributed communications networks 30 years ago, we have seen computing, communications, and telecommunications converge into today's Internet of global reach and empowerment. The complexity of the evolved infrastructure demands a cautious approach to policy and practice.

Standards for access, content, procedures, and security must be carefully thought through in order to ensure that the anytime, anywhere paradigm of the new learning environment is preserved and nurtured. Perhaps the greatest challenge of all is culturally adapting to the new paradigm and the new capabilities. The DoD must re-engineer the way it learns and, by extension, the way it does business.

The ADL Team in DoD is working diligently to help the entire department maximize the benefits from adapting learning technologies. They have established a Total Force Advanced Distributed Learning Action Team, which meets monthly to document and coordinate learning initiatives throughout the military departments and defense agencies. In conjunction with the Department of Labor and the Institute for Defense Analyses, a nonprofit think tank, they have established an ADL co-laboratory to bring government, industry and academia together to test commercial product offerings against learning requirements. A *sine qua non* for the success of the ADL Initiative is the implementation of standards and guidelines that have been drafted via collaboration between the department and major standards development organizations, including the Instructional Management System, Aviation Industry Computer-based Technology Committee, and Institute of Electrical and Electronic Engineering—with DoD co-sponsorship—to ensure cost-effective sharing and reuse of content. In addition, the Director for Defense Research and Engineering, military department laboratories, and several academic institutions have begun significant efforts to research how we can use such technologies to learn and to find ways to accelerate the process.

The ultimate ADL Initiative goal is to deliver effective learning where and when it is needed by the men and women of the Department of Defense. Doing so is essential to making them ready to respond to an uncertain international security environment that demands that they make more deployments, more rapidly, for more complex missions, than ever before. For more information about the ADL Initiative, visit www.adlnet.org◆

About the Author

As Director for Readiness and Training Policy and Programs in the Office of the Secretary of Defense, **Mike Parmentier** is the OSD focal point for for all DoD training policies and programs. He guides and oversees DoD's training activities and budgets, including infrastructure (base realignment and closure, land withdrawal, ranges), learning technologies (distributed learning, distributed simulation, range instrumentation), exercises, equipment acquisition, training readiness reporting, and research and development. Parmientier leads several training committees, serves as Executive Secretary for the Defense Science Board Task Force on Military Training and Education, and represents the DoD on the President's Task Force for Federal Training Technology.

Cognitive Readiness and Advanced Distributed Learning

This article discusses the Cognitive Readiness focus area, and how the Department of Defense (DoD) Advanced Distributed Learning initiative will provide crucial and timely near- and mid-term enabling capability.

Ensuring that the warfighters of today and tomorrow have superior, affordable technology to support their missions is the DoD Science and Technology objective. To that end, five key focus areas have been identified where DoD corporate investment in multidisciplinary experiments and demonstrations could leverage existing programs that address:

1. Chemical and biological defense
2. Hardened and deeply buried targets
3. Smart Sensor Web
4. Information assurance
5. Cognitive Readiness

The Cognitive Readiness vision is to optimize the human contribution to joint warfighting, and achieve the revolutionary war-winning capability articulated in DoD's Joint Vision 2010. In operational terms, achieving Cognitive Readiness ensures that:

- The warfighter is *mentally prepared* for accomplishing the mission.
- The warfighter is *performing* at his/her optimal performance level.
- The tools and techniques for *preparing* the warfighter are the most effective and affordable.
- The tools and techniques the warfighter *uses* are the most effective and affordable.

Fundamentally then, Cognitive Readiness focuses our science and technology efforts on addressing the critical need for increased capability and adaptability from the human component of weapon systems in a progressively more complex, dynamic, and resource-limited environment. Although there is now a lessened risk of facing a single massive threat, the joint warfighter is challenged by the potential of simultaneous, multiple, geographically separate, high- or low-intensity conflicts, as well as peacekeeping, counterterrorism, and disaster support missions—all with tightly constrained operations and acquisition resources.

These post-Cold War challenges have compelled adaptive adjustments by the services. Today's forces are deployed more frequently and for longer periods of time, often in urban situations where the individual warrior is the weapons platform of choice. These changes in military context and capability, as exemplified in Figure 1, even now routinely stress the capacities of individuals and teams across a spectrum of tasks and operations.

- High personnel and operational tempos (PERSTEMPO, OPTEMPO)
- Flying time issues
- Safety
- Operating space availability
- Dwindling exercise $$
- Few joint training opportunities
- Complex rules of engagement
- Classified capabilities
- Restricted weapons/electronic warfare envelopes
- Environmental concerns

Figure 1. *A Sampling of Training Constraints/Challenges*

The Cognitive Readiness focus area employs a multidisciplinary systems approach to address these performance challenges. As shown in Figure 2, it draws upon the human systems and biomedical areas of the DoD science and technology programs, integrating scientific and applied contributions from health, psychology, sociology, and human factors engineering fields.

- **Education -** Teaching generalized knowledge (cognitive)
- **Training -** Teaching specific tasks/ knowledge (cognitive, psychomotor)
- **Human Systems Integration-** Modifying system inputs/outputs and environment to optimize human performance and understanding cognitive processes
- **Sociology -** Family, group, and culturally defined issues
- **Health -** Mental acuity, physiological wellness

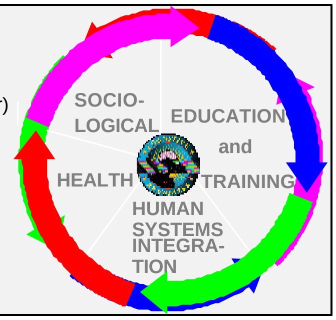


Figure 2. *Cognitive Readiness: A Multidisciplinary Research Program*

Although a variety of research efforts develop and apply technologies toward improving human performance, it is seldom that resulting demonstrations and deliverables leverage or generalize beyond specific applications, missions, or components. Cognitive Readiness provides the integrating mechanisms for coordinating, overseeing, and supplementing focused research across the DoD, industry, and academia. An initial high-payoff area is the technologies necessary for the DoD's education and training missions.

We are greatly interested in achieving the capability to train and educate our military and civilian workforce, anytime and anywhere it is required, with systems tailored to individual needs. Today's tremendous advances in information technology put us on the verge of an enabling environment for Advanced Distributed Learning (ADL). In the end-state, ADL hardware and software must have the characteristics of accessibility, interoperability, durability, reusability, and cost effectiveness. While communications and computing technologies are on an evolutionary track to accommodate ADL accessibility, technologies to accommodate the other characteristics are severely limited. Most computer-based and Web-enabled training and education systems are static, single-point answers to a single need. They are difficult to scale or adapt to large and diverse learner communities. Content and "courseware" are rigid, not designed for reuse. Knowledge management and search tools are mostly inadequate for all but the most superficial uses. Many educational technologies are high in cost, low in reliability, and difficult to adapt to special usability needs. A review of software industry trends indicates that many companies now believe that an object-based approach will provide the basis for platform neutrality and software reusability needed for the large-scale development and dissemination of powerful and cost-effective learning content. To achieve the ADL objective, it is important to initiate research that enables platform neutrality and software reusability.

A recent front-end assessment and workshop identified key components for a research agenda to achieve a robust, national ADL capability by the end of the decade. Participants represented a cross-section from the services, government, industry and academia. These people were recognized experts in areas that included education, training, curriculum development, software engineering, hardware engineering, educational research, cognitive, and behavioral science. The four key research areas that were identified as necessary to enable the ADL vision of a readily available instructional environment to support anytime, anyplace, anyone, anything learning were:

- Intelligent Computer-Aided Instruction.
- Authoring Tools.
- Distributed Simulations.
- Dynamic Learning Management.

We are embarked upon establishing a comprehensive, multidisciplinary research program to address these fundamental research areas.

This program will serve as the basis for providing accessible, tailorable, and affordable training and education to military and DoD civilian employees through advanced distributed learning. Underlying achieving the ADL vision by the end of the next decade is the requirement for a supporting hardware and software infrastructure. ADL will be enabled by an open, evolving learning technology environment based on a ubiquitous, distributed infrastructure with interoperability of components and learners across a multitude of bounds (e.g., spatial, temporal, organizational, technological). The DoD software engineering community will be a key to this strategic initiative's success.

In sum, Cognitive Readiness is both a critical component and a criterion for the DoD's Science and Technoogy strategy for achieving the national defense capability articulated in Joint Vision 2010. It forces emphasis on achieving national advantage through optimizing the capability and employment of our people—our nations greatest asset—for peace, as well as war. ADL, in turn, provides a supportive strategy that will contribute to the achievement of cognitive readiness. Accelerated and sustained S&T investment in ADL should yield near- and mid-term dividends that will dramatically enhance our forces' cognitive readiness. The challenge is daunting, not only because of the infrastructure and human resource S&T requirements but also because of the organizational, cultural, and security issues that must be addressed. Clearly, realizing the promise and the potential of ADL will be dependent upon a shared vision and a nation-wide, multidisciplinary team effort.◆

About the Authors

Dr. Delores M. Etter is the Deputy Under Secretary of Defense for Science and Technology. She is responsible for Defense Science and Technology strategic planning, budget allocation, and program execution and evaluation. Additionally, she ensures that the National Defense objectives are met by the $8 billion per year DoD Science and Technology Program. Etter's research interests are in adaptive signal processing, speech and speaker recognition, digital filter design, and software engineering. She has written a number of textbooks on computer languages and software engineering. Her educational interests include the development of collaborative experiments in virtual teaming of students using the Internet.

Dr. Robert E. Foster is the Director for BioSystems in the Office of the Deputy Under Secretary of Defense for Science and Technology. Foster is responsible for coordination and oversight of the DoD's biomedical, human systems, training, counterterrorism, and environmental quality science and technology programs. Foster is also responsible for oversight of the department's animal and human use regulatory affairs programs. His professional scientific background includes neuroscience (anatomy and neurophysiology), toxicology, and human sensory system psychology with multiple government and peer-reviewed publications.

Cmdr. Timothy P. Steele is a Navy research psychologist serving as the Assistant Director for Human Systems in the Office of the Deputy Under Secretary of Defense for Science and Technology. He serves as principal advisor and action officer for the DoD's Human Systems S&T Program, which includes the subareas of information display and performance enhancement, personnel performance and training, design integration and supportability, and warrior protection and sustainment.

Sorry to See You Go, Joe

CROSSTALK bids farewell to Lt. Col. Stanley Joseph Jarzombek Jr. this month. After nearly four years as the sponsor of CROSSTALK, Jarzombek is retiring as the Director of the Computer Resources Support Improvement Program (CRSIP [formerly ESIP]).

As the CRSIP director, he headed the Air Force software technology and software knowledge management initiatives and provided $16 million a year infrastructure to support technologies of Air Force organizations. His many tasks included sponsoring and directing CRSIP efforts in the Software Technology Support Center. In that position, he oversaw technology information services in addition to this journal: online web information services, and the Software Technology Conference, the Department of Defense's premier software conference that draws 3,500 attendees annually.

With Lt. Col. Jarzombek at the helm since July 1996, CROSSTALK has evolved from a black-and-white digest to today's journal with its signature purple signifying tri-service coverage of software issues, professionally created cover art, and a more reader friendly and usable format. Thanks to his support, readers will soon see another major publishing step: a full-color cover.

His software involvement and interest are far-reaching. At CRSIP he sponsored Air Force Research Lab software technology research, development, and evaluation services to provide solutions for software support environments and migration of software legacy systems. He directed tasks and studies with SEI on best practices, SPI efforts, and documentation. He provided corporate direction for software readiness programs through sponsorship of SEPGs in software divisions within AFMC centers. He directed the upgrade of the Air Force's network of software control centers. His other activities include membership on Web editorial boards to provide coordinated software technology initiatives, policies and practices, and service on the DoD-sponsored CMMI® Product Development Team. Thanks, Lt. Col. Jarzombek, for your valuable support!

Software Engineering Degree Programs

Today, software engineering is a separate degree program in academia. The implications for the Department of Defense (DoD) are that the new graduate, once on the job, is going to be better prepared to contribute as a technical team member with appropriate experience and knowledge. This article outlines some of the changes that are taking place and briefly describes accreditation direction for such programs. It is anticipated that the new graduate will be better prepared to immediately contribute to software engineering organizational goals.

After more than 25 years in the defense software business and now in academia, this author is pleased to report that many academic institutions are beginning to offer, or are considering, undergraduate and graduate software engineering (SE) degree programs in addition to existing computer science programs. This is true of the author's institution where approval for its new SE bachelor's degree program is expected this year.

These emerging degree programs are more directed at the practical than the theoretical. Shari Pfleeger [1] synopsizes the difference between computer science and a software engineering focus very well in her recent text on software engineering where she writes, "*We can concentrate on the computers and programming languages themselves, or we can view them as tools to be used in designing and implementing a solution to a problem. Software engineering takes the latter view . . . .*"

The Institute of Electrical and Electronics Engineers Standard 610.12 defines software engineering as "*the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software*."

In all software engineering programs this author is familiar with, computer science, computer engineering, and industrial engineering are supporting fields. The combination of these disciplines and the focus on practical application will lead to a different kind of graduate than those coming out of more theoretical programs today. This change in emphasis is likely to meet with approval on the part of the DoD and industry who hire our graduates and expect them to be technically proficient and immediately productive.

This article outlines the essence of these new programs and provides some insight into a typical curriculum associated with them—a curriculum that should better meet the current needs of the software engineering community and address the software engineering crisis so often referred to in today's literature [2, 3].

Computer science programs have been around a long time and for the most part have included software engineering course offerings as required or elective coursework. Business information systems degrees also have been available for some time—many dating to the mid-1960s or early 1970s. Computer engineering programs also exist with well-established curricula. With this wide variety of related programs one might question the need to introduce a degree specifically in software engineering. It helps to look at it from two perspectives—first reviewing the focus of each of the degree programs mentioned above and secondly, by considering the state of our software infrastructure today and how existing programs might be expected to contribute to solving problems. These programs are characterized later in this article.

Computer science programs generally focus on the principles, applications, and technologies of computing with an emphasis on data, data structures, algorithms, computer architectures, and theory of computing. Computer science also includes the fields of artificial intelligence, graphics and/or scientific visualization, computer language design, structure, and translation. Problem solving and design methodology is often included. Such programs are largely, but not entirely, theoretical. Business information systems programs tend to focus the student more on the management aspects of providing computing services to an enterprise—costing, accounting, economic impact, systems analysis procedures, and business data processing. One university's course catalogue description for such a program [4] includes the statement " . . . must have a broad background and understanding of the business environment, including such topics as accounting, economics, law, management, production, marketing, finance, and communications."

Clearly, such information systems programs are directed at producing a graduate who serves an important role in the operations, planning, and maintenance functions associated with an enterprise's supporting computer infrastructure —but not one prepared to lead technical efforts in building large-scale, software-intensive systems. Computer engineering programs grew out of electrical engineering applied to the computing domain. Such programs tend to be more oriented to the hardware aspects of computing, with emphasis on circuit design and a strong complement of software courses to include algorithms and programming skills. The graduating computer engineer has a strong background in mathematics, circuit theory, digital devices, architecture, and chip design, as well as basic computer science that includes programming, data structures, networks, and systems theory.

Each of these three programs in any academic institution is comprised of somewhere around 120-140 semester hours (where one class, on average, is worth three credit hours). Once a student finishes all required core courses, about 40 percent of the course work is accomplished in the specific major field of study. It becomes apparent that universities today have little expansion ability in their programs, and adding a course usually means that something else must go. It would seem that encouraging any of the degree programs listed above to focus more on the tools and techniques of software engineering can only come at the expense of other important topics that form the essence of the degree.

Software Engineering

This is an area of interest that has been with us for around four decades. The early years involved very little engineering, comprised mostly of tools and procedures used in programming. About the mid-1960s we began to notice a significant problem in living with the software we had created—it was largely difficult and expensive to maintain. With about 450 different languages documented in the

DoD at that time, coupled with few programming standards that existed or were followed, today's software crises began to appear. We spent the next two decades researching ways to improve and measure our processes, reduce costs and ease the maintenance issue, correctly build software and to prove we did so, and incorporate rigorous process descriptions. We approached these problems largely through our computer science programs, which usually included a software engineering component, and a certain complement of skills associated with modern programming languages and procedures.

The emphasis of such programs was —and remains—primarily theoretical with a strong emphasis on improving our techniques and processes for tomorrow's development environments and systems. Today, software engineers are needed who have more depth in current accepted procedures, processes, tools, measures/metrics, and a quality focus. To meet this need, over the past few years the academic community began to agree that software engineering has matured to the point that it is now a credible separate degree program with a community whose needs are unmet by other programs. The accrediting body for engineering degree programs, the Accreditation Board for Engineering and Technology, adopted criteria for software engineering. Programs to meet industry's needs are beginning to emerge at the undergraduate level.

The new graduate, according to the desires expressed in the accreditation criteria (www.computer.org/tab/swecc/accred.html) will be able to demonstrate an ability to analyze, design, verify, validate, implement, and maintain software systems using appropriate processes, models, and metrics in software development. This graduate also is expected to possess necessary team and communication skills to function in a typical software development environment. Those of us that have worked in these "typical software development environments" will welcome this change of emphasis.

It has been this author's experience that graduates join industry with good technical skills but lack the office social skills so important to overall corporate success. We seem to have too many assignments in academia that all start with the instructions, " Do your own work, do not get help from any outside source, do not give any help to others working on this assignment, do not talk to others about this assignment . . ." and so forth.

When our students reach their first employers they are put onto a team, expected to cooperate with others, share their work, help each other, communicate, etc. We reverse the paradigm and many find it difficult to adapt. Our emerging software engineering degree programs should help modify this approach by introducing more team-oriented projects, focusing on deliveries that require documentation built to professional standards, and requiring students to think in terms of an entire project life cycle rather than on a specific program due before 3 p.m. Friday. The accreditation criteria also suggests a curriculum that should include approximately equal segments in software engineering, computer science, and supporting areas, with the total courses covering about three-fourths of the total academic program. This will undoubtedly allow our graduates to spend the time needed to learn corporate practices, quality, program management principles, and the process needed to deliver and maintain large-scale software-intensive systems.

Software Engineering Accreditation Expectations

It is the accrediting bodies in academia that insure certain educational goals are met and that they can be met in a particular institution. Periodically, degree programs are assessed by these accrediting bodies and are approved or denied accreditation. Software engineering criteria contain certain requirements that are most certain to find favor within the large DoD community of employers outlined below:

Faculty Requirements: A key expectation is that faculty who teach core software engineering courses should have substantial practical software engineering experience. Faculty must also be able to interact effectively with software practitioners. The implication appears to be that the faculty should have the ability and experience to teach practical skills necessary to enter the software engineering work force and to develop collaborations with industry. Many academic departments today promote the faculty's ability to obtain research grants from organizations like the National Science Foundation (NSF) or the Defense Advanced Research Projects Agency (DARPA), but do not value industrial or government agency collaborations as highly. Often NSF or DARPA grants, for instance, count heavily toward the tenure process that a faculty member faces, whereas industrial collaborations may not —removing the motivation to enter into such arrangements. The software engineering faculty member may, in future programs, be encouraged to reverse this trend.

Curriculum: The criteria specifically states that a central theme "is to engender an engineering discipline in students, enabling them to define and use processes, models, and metrics in software and system development." Curriculum guidance includes theory and practice, but the emphasis is on practice. Programs must include all aspects of software development and maintenance and provide for experience in a realistic team environment. Written and oral communication skill courses are promoted as appropriate supporting courses. Courses such as development and maintenance, requirements analysis, architecture and design, testing, and quality assurance are required. Who among us would not want such a graduate? These changes will certainly bring more valuable employees to the workplace and will allow them to become immediately productive.

Laboratory and Computing Resources: Substantial laboratory and computing resources are expected in these programs, both for student exposure and support as well as to support the research of the faculty members. Such laboratories are to be populated with a wide variety of tools, computing facilities, operating systems, and commercial products. Interestingly, the accreditation criterion also suggests meeting space to support team projects, an essential aspect of software engineering training. Laboratory support staff is recommended, although in most academic institutions, it is difficult to find funding for permanent laboratory support staff.

Institutional Support: Support from the university administration is essential for software engineering programs. The obvious support needed is opportunity for faculty to stay current, good library support, resources to build strong programs, and support for collaboration with industry and government software engineering organizations. Less obvious areas of support that would likely be considered helpful include tenure credit for working with industry, industrial affiliation agreements between the university or software engineering program, industry/government organizations, and support for distance learning technologies.

Support for distance learning technologies would assist in electronically including working adults in software engineering organizations as members of software engineering classes on campus.

Industrial affiliation agreements would lead to exchange opportunities, student co-op programs, internships, and work collaborations.

Summary

Change is on the horizon for those concerned with addressing the need for software engineers in the future. This is a welcome change, heralding the maturing of our discipline in the eyes of others. Several undergraduate software engineering programs exist today and more are planned.

The need is clear. This nation has an aging software infrastructure that must be maintained and replaced. We have begun to develop the necessary rigor in our profession to teach a common body of knowledge (see www.swebok.org for more on this subject).

At least one state—Texas—has begun a process of licensing software engineers as professionals, and we have learned hard lessons over the years regarding the need for process and how to measure it. Never has there been a more exciting time to teach software engineering and research improvements in the state of the practice.

Those in the software engineering profession can look forward to the new, improved graduate beginning to appear on the scene in the immediate future. For more information on the planned program at Mississippi State University, please contact the author directly. For more information on software engineering programs visit the following web sites:

www.swecc.org
faculty.db.erau.edu/hilburn/se-educ/
www.lrgl.uqam.ca/publications/pdf/365.pdf
www.cs.utexas.edu/users/ethics/professional.html
www.main.org/peboard/softw.htm ◆

References

1. Pfleeger, Shari, *Software Engineering Theory and Practice*, Prentice Hall, 1998
2. Gibbs, W.W., "Software's Chronic Crisis," *Scientific American*, Sept. 1994, pp. 86-95.
3. Royce, W.W., Current Problems, *Aerospace Software Engineering: A Collection of Concepts*, C. Anderson and M. Dorfman, eds., American Institute of Aeronautics, Inc., Washington, D.C., 1991.
4. Mississippi State University, *The Bulletin of the Mississippi State University 1998-1999*, P.O. Box 5268, Mississippi State, Miss. 39762

About the Author

Rayford B. Vaughn Jr. spent 26 years in the Army as a software engineer. His key assignments included commander of the Army's Information Systems Software Center headquartered at Fort Belvoir, Va. and the first Director of the Pentagon Single Agency Manager for Information Technology Services. Upon retirement as a colonel in 1995, he joined Electronic Data Systems (EDS) Military Systems as vice president of Defense Information Systems Agency (DISA) Integration Services, where he was responsible for all EDS contracts issued by DISA. In October 1997, he accepted a position as an associate professor of computer science at Mississippi State University, where he teaches software engineering and computer security courses at the graduate and undergraduate level. He is a past contributor to CROSSTALK and a frequent speaker at the DoD Software Technology Conference.

Department of Computer Science
PO Box 9637
Mississippi State, Miss. 39762
Voice: 662-325-7450
Fax: 662-325-8997
E-mail: vaughn@cs.msstate.edu
Internet: http://www.cs.msstate.edu

Education and Training Web Sites

http://rbse.jsc.nasa.gov/ricis/research-area/et/et.html
Links cover, but are not limited to, software engineering and Ada training; research in intelligent tutoring systems for knowledge-poor domains; software engineering and training implementation research; hypermedia tools for building technical training systems, information systems research center; continued research and development with the microcomputer intelligence for technical training concept; and establishing an integrated education program in software engineering and computer engineering training, phase one.

http://www.isse.gmu.edu/~ofut/rsrch/see.html
Visit this site for papers on "Maintaining Knowledge Currency in the 21st Century, "A Five-Year Perspective on Software Engineering Graduate Programs at George Mason University," "Integrating Research, Reuse, and Integration into Software Engineering Courses," and "Anatomy of a Software Engineering Project."

http://www.asafm.army.mil/profdev/acteds.htm
This site is by the Army Civilian Training Education and Development System. Those viewing it need Adobe Acrobat Reader or another application capable of reading .pdf files. This site provides systematic training and development for careerists from intern to senior managerial and executive levels. The plan outlines sequential and progressive training in functional specialties, leadership, supervision, and managerial development.

http://www.seminarfinder.com/search/
This is used to find seminars, Web-based training programs and continuing education. Visitors to this site can browse by city or topic. There also are links to training, and affiliate programs.

http://www.iee.org.uk/PG/S5
Professional Group S5 of the Institution of Electrical Engineers provides this site on the processes of education and training in the context of electrical, electronic, manufacturing, and software engineering. It covers, but is not limited to, such things as methods of attracting young people into the profession and continuing professional development, special educational requirements for identifiable sectors of industry and related industrial training programs, and education and training aspects of specific subject areas.

Industry/University Collaborations
Different Perspectives Heighten Mutual Opportunities

In this paper, we present the results of a survey by the SEI Working Group on Software Engineering Education and Training of formal industry/university collaborations. The purpose of these collaborations is to meet the software engineering education and training needs of adult learners through joint ventures such as graduate programs and professional development activities. The Working Group drew on the extensive experience of industry and university collaboration participants to help answer practical questions about the benefits of collaboration, the collaboration process, successful collaboration administration and programming, and lessons learned.

Universities and industry organizations have traditionally maintained *informal* ways of working together, including student internships, faculty exchanges, and industry capstone projects to complete a degree program. A more recent phenomenon is the *formal* collaboration between a university (or group of universities) and an industry organization (or group of organizations). The purpose is to meet the critical software engineering education and training needs of adult learners through joint ventures such as graduate programs (degree and certificate) and professional development activities (customized classes, seminars, forums, and conferences).

In September 1995, the Software Engineering Institute (SEI) began tracking this phenomenon in its annual *Directory of Industry and University Collaborations with a Focus on Software Engineering Education and Training* [1]. There are now 23 directory entries, representing collaborations formed in the United States, Canada, and Australia. Several of these pioneer efforts are now six to 10 years old.

There is some previously published literature on the topic. In 1994, Neal Coulter and Jim Dammann (1994) published the results of a successful collaboration at Florida Atlantic University [2]. The November/December 1997 *IEEE Software Special Issue on Software Engineering Education and Training* contained the article, "Industry/University Collaboration: Closing the Gap Between Industry and Academia," which described a model for collaborations and featured a close look at three of these efforts [3]. However, industry/university collaborations for software engineering education and training remain a mystery to many in academia and industry. Commonly asked questions are practical queries such as the following:

- Why collaborate? What are the benefits?
- How would we start, and then operate, a collaboration?
- What makes a collaboration successful?

In other words, what makes an industry/university collaboration "tick"?

The SEI Working Group on Software Engineering Education and Training contacted collaboration practitioners to help answer these practical questions [see sidebar on p. 15]. This paper provides some preliminary answers. It is offered as a service to the software engineering education and training community to assist anyone interested in forming a new collaboration or improving an existing collaboration.

Survey Methodology

In December 1997, the SEI Working Group surveyed both academic and industry representatives of the 23 collaborations documented in the 1997 Directory, as well as collaborations from Auburn University and Lockheed Martin Corp. The 25-question survey requested:

1. Demographic information.
2. Information on collaboration goals and measures.
3. Overall collaboration process information.
4. Individual collaboration activities and results.
5. Lessons learned.

Fourteen responded. Participating universities and their industry partners are listed below:

- American University with Center for Systems Management
- Applied Information Management Institute at Creighton University with First National Bank of Omaha
- Auburn University [in development]
- Boston University Corporate Education Center with Affiliated Computer Services (ACS) Government Solutions Group (formerly Computer Data Systems Inc.)
- Embry-Riddle Aeronautical University with ACS Government Solutions Group (formerly Computer Data Systems Inc.)
- Florida Atlantic University with Allied Signal, CITRIX, Encore Computer Corporation, Harris, IBM, Motorola, Sensormatic, Siemens Telecom, and United Technologies
- Lockheed Martin Tactical Defense Systems with University of Akron
- Lockheed Martin Idaho Technologies Company with Boise State University, Idaho State University, Montana State University, University of Idaho, and Utah State University
- Software Engineering Forum for Training, California State University/Long Beach (SEFT/CSULB) with The Boeing Co., Northrop Grumman, and TRW
- Software Engineering Research Centre of Royal Melbourne Institute of Technology and University of Melbourne with Ericsson Australia
- Software Quality Institute, University of Texas at Austin with 28 representatives from industry and government
- University of California Santa Cruz with Santa Cruz Operation, Seagate Technologies, and Thuridian
- Texas Tech University with Raytheon Co.
- University of Maryland University College with ACS Government Solutions Group (formerly Computer Data Systems Inc.)

From the respondents, the Working Group selected the following for follow-up phone interviews:

1. American University/Center for Systems Management
2. Florida Atlantic University
3. Software Engineering Forum for Training, California State University/Long Beach (SEFT/CSULB)
4. Texas Tech University

Summary of Overall Survey Data

Demographic Information

Responding collaborations generally have been active for one to five years; however, there are instances of collaborations operating for 10 years or more. Table 1 depicts demographic information from the 14 collaborations responding to the survey. The types of software engineering education and training services provided through the collaborations are documented in the column on the right. Graduate programs denote master's degree or graduate certificate programs; professional development signifies noncredit courses, seminars, and conferences.

Why form an industry/university collaboration?

There are many reasons for forming an industry/university collaboration. These reasons include fulfilling an organization's education mission, accessing education and training resources, gaining competitive advantage, addressing business growth, achieving cost savings, enhancing organizational reputation, increasing revenue, accessing research and tool resources, and providing a staffing source.

Fulfilling the organization's educational mission was selected most often as the respondents' first priority (seven ratings), followed by business growth (three ratings), access to education and training resources (two ratings), and staffing source (two ratings).

What do industry/university collaborations offer?

Collaborations offer a variety of software engineering education and training activities, including classes, seminars, conferences, workshops, and certificate and degree programs. These activities are typically held on location, either at the university or industry site, although circumstances may vary. The survey revealed a healthy number of participants is served each year by collaboration activities. The surveyed collaborations reported the attendance shown in Table 2.

Courses cover software management topics, integrated product teams, electrical engineering topics, principles of software development, system engineering applications and practices, and simulation models for operations analysis. The various

Collaboration Name	University(ies)	Industry Partner(s)	Location	Types of Services
AU/CSM	American University	Center for Systems Management	Washington, DC	Professional development and graduate program
Applied Information Institute (AIM)	Creighton University	Member companies	Omaha, Neb.	Professional development
Strategic Occupation Alliance Resources (SOAR)	Auburn University	[in development]	Auburn University, Ala.	Professional development
	Boston University Corporate Education Center	ACS (Affiliated Computer Services) Government Solutions Group	Boston, Mass.	Professional development
	Embry-Riddle Aeronautical University	ACS Government Solutions Group	Daytona Beach, Fla.	Professional development
	Florida Atlantic University	Allied Signal CITRIX Encore Computer Corp. Harris IBM Motorola Sensormatic Siemens Telecom United Technologies	Boca Raton, Fla.	Professional development and graduate program
	University of Akron	Lockheed Martin Tactical Defense Systems	Akron, Ohio	Professional development and graduate program
	Boise State University (Boise, Idaho) Idaho State University (Pocatello, Idaho) Montana State University (Bozeman, Mont.) University of Idaho (Moscow, Idaho) Utah State University (Logan, Utah)	Lockheed Martin Idaho Technologies Company	Idaho Falls, Idaho	Graduate program
Software Engineering Forum for Training (SEFT)	California State University, Long Beach	The Boeing Co. Northrup Grumman Corp. TRW	Costa Mesa, Calif.	Professional development
Software Engineering Research Centre (SERC)	Royal Melbourne Institute of Technology, University of Melbourne	Ericsson of Australia	Melbourne, Australia	Professional development
Software Quality Institute	University of Texas at Austin	28 representatives from industry and government	Austin, Texas	Professional development
	University of California Santa Cruz	Santa Cruz Operation, Seagate Technologies, and Thuridian	Santa Cruz, Calif.	Professional development and graduate program
Systems Engineering Master's Program – with Software Engineering component	Texas Tech University	Raytheon Co.	Lubbock, Texas	Graduate program
	University of Maryland University College	ACS Government Solutions Group	College Park, Md.	Professional development

Table 1. *Summary of Overall Survey Data*

formats of content delivery include practical hands-on seminars, use of distance learning technologies, and formal classroom instruction. For more detailed course offerings, refer to the *Directory of Industry and University Collaborations with a Focus on Software Engineering Education and Training* at http://www.sei.cmu.edu/publications/documents/97.reports/97sr018/97sr018title.htm.

Software Engineering Education and Training Activity	Attendance (Annual total for responding collaborations)
Courses	593
Seminars	496
Conferences	1,500
Workshops	353
Certificate Programs	225
Degree Programs	337
Other: Software Process Improvement Network	1,000

Table 2. Collaboration Attendance Records

How does a collaboration work?

Collaborations must be initiated. This is crucial and requires a strong champion on both sides. Of all the respondents, 44 percent indicated that they sought out a new contact with industry, university, or government to establish the collaboration, while 26 percent expanded an existing relationship to get started. The remaining collaborations were initiated by faculty who worked for both partners or through a Software Process Improvement Network contact.

The collaborations are funded by multiple sources, including annual membership fees, activity-based fees, grants, or contracts. A full 90 percent of all collaborations surveyed document their collaboration goals, while 70 percent have a formal agreement such as a charter, memorandum of understanding, or contract. Of those surveyed, 79 percent use consensus among collaboration participants to make operational decisions. They use varied organizational structures such as advisory boards, boards of directors, or program managers to steer the collaboration's activities. Two-thirds of the collaborations surveyed employ between one and three paid staff members to support collaboration activities. Figure 1 is the basic collaboration process.

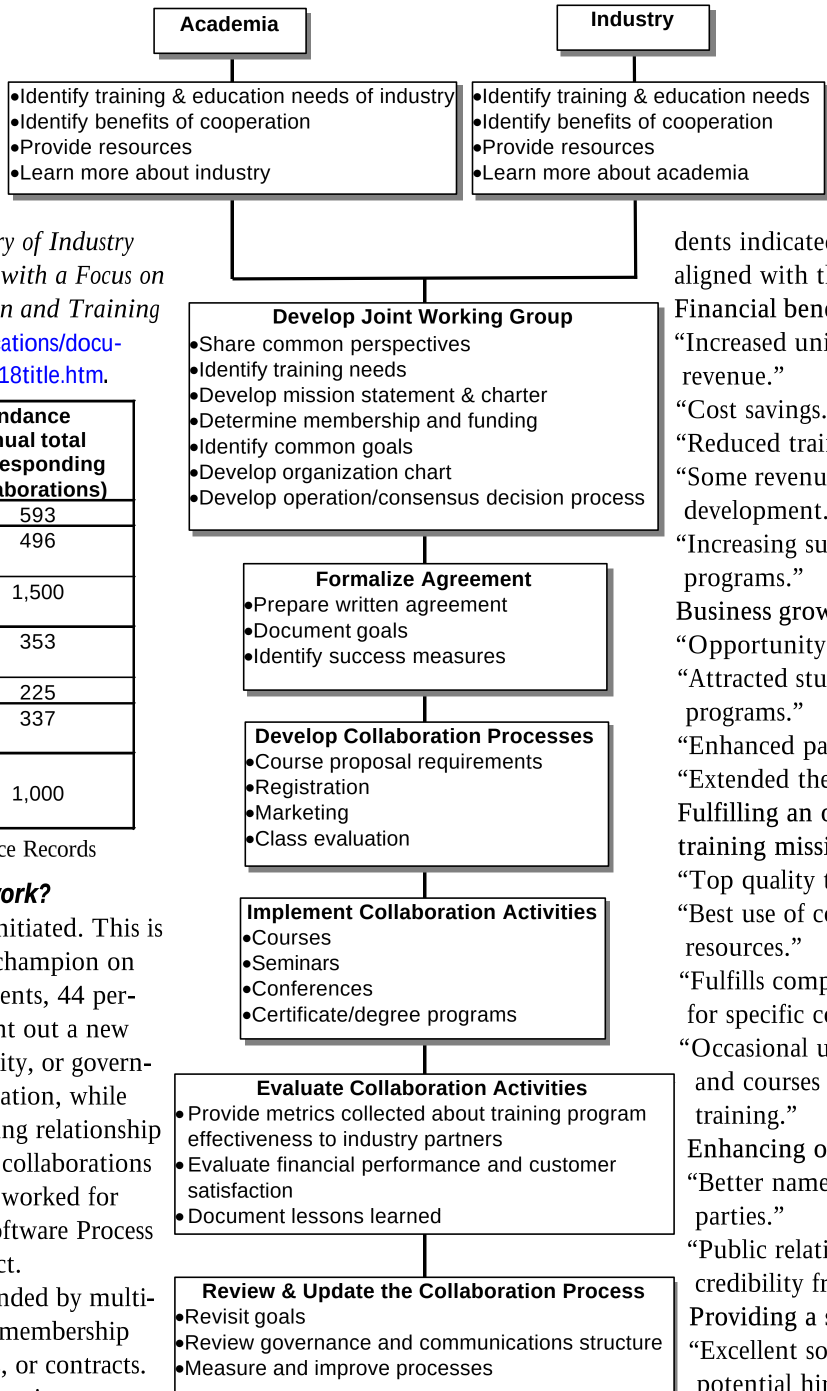


Figure 1. *Academia and Industry Collaboration Process*

What are some benefits of collaborations?

Survey respondents were asked to list benefits realized from their collaborations. The list indicated collaborations could realize their goals and sometimes benefit in ways not previously expected.

Quotes from respondents indicated the following benefits aligned with their stated goals.

Financial benefits

"Increased university and partner revenue."

"Cost savings."

"Reduced training costs per employee."

"Some revenue for program development."

"Increasing support for research programs."

Business growth

"Opportunity for follow-on business."

"Attracted students for degree programs."

"Enhanced partner's marketing."

"Extended the reach of the university."

Fulfilling an organization's education/training mission

"Top quality training."

"Best use of company's training resources."

"Fulfills company training requirements for specific courses."

"Occasional use of academic knowledge and courses to supplement industry training."

Enhancing organizational reputation

"Better name recognition for both parties."

"Public relations benefit and local credibility from the partnership."

Providing a staffing source

"Excellent sources of interns and potential hires."

Survey respondents also reported important collaboration benefits they did not specifically set out to achieve, such as:

"Knowledge of workings of the opposite sector."

"Sharing of knowledge and experience of member companies."

"Insight into member companies' training programs, issues, problems, and experiments."

"A community of practitioners able to share their expertise as a community of learners rather than as competitors."

"Sharing knowledge and use of emerging technology (distance learning and Web-based training)."

Results of Follow-Up Interviews

After gathering initial data, the SEI Working Group looked at the collaborations that reported having documented processes and had been active for two years or more. The Working Group conducted follow-up phone interviews with representatives from four of these collaborations: Texas Tech University, Florida Atlantic University, Software Engineering Forum for Training, California State University/Long Beach (SEFT/CSULB), and American University/Center for Systems Management (AU/CSM). Although the collaborations interviewed represented different collaboration models and missions, the data they shared with the Working Group provide useful details on successful collaboration administration and programming.

Collaboration Models and Missions

Single university/single industry collaboration. Texas Tech in Lubbock, Texas provides a master's degree in systems engineering with software engineering component to Raytheon Co. employees.

Single university/multi-industry collaboration. SEFT/CSULB provides tailored training in software process improvement and management practices to employees of three member companies from the aerospace industry—The Boeing Co., Northrop Grumman Corp., and TRW. Florida Atlantic in Boca Raton, Fla. collaborates to deliver graduate software engineering courses to employees of nine research and development firms with headquarters or major plants in southeast Florida: Allied Signal, CITRIX, Encore Computer Corp., Harris, IBM, Motorola, Sensormatic, Siemens Telecom, and United Technologies.

University/consultant collaboration. American University, Washington, D.C., and the CSM, San Jose, Calif., jointly market and deliver a 15-credit Graduate Certificate Program in systems and project management, with short courses in these topics.

All four collaborations originated from existing industry/university/government alliances. They share similar missions (e.g., enhanced institutional reputation, access to affordable training resources, and a source of revenue). Also, SEFT/CSULB and Florida Atlantic view their collaboration as a vehicle for serving the software engineering education and training needs of the regional community. Florida Atlantic established a goal to obtain access to research, training, and educational resources from its industry partners.

Successful Administration

Industry/university collaborations have a myriad of organizational formats due to the variety of projects encountered and the organizations involved. For the most part, the four collaborations interviewed are formal arrangements with Florida Atlantic the exception. However, Florida Atlantic's informal arrangement involves committee activity, and—as with all of those interviewed—there is mutual planning and review with industry partners. Also, all four organize activities, make operational decisions, and administer their collaborations through joint industry/university responsibilities. Additional joint responsibility of administrative functions was found primarily in marketing activities of the collaborations (e.g., American University/CSM and SEFT/CSULB), in contacting new organizations, expanding existing relationships, or working through industry/professional associations for new members to add to or replace members in their collaborations.

Collaboration	Program(s)	Program Format(s)	Site(s)
Texas Tech University	Master's degree	Classroom Distance education delivery includes video, lecture, and Web-based instruction	University sites
SEFT/CSULB	Professional development	Classroom	Client and university sites
FAU	Master's degree and professional development	Classroom Videotape and live broadcast over Florida Engineering Educational Delivery System	Client and university sites
AU/CSM	Graduate certificate program	Classroom	Client and university sites

Table 3. *Software Engineering and Training Planning*

Two governance models were noted. The first was joint university-industry input and evaluation (Texas Tech and Florida Atlantic). American University was similar, as it included its industry partner CSM in program administration planning and decisions. A different governance structure was found in the SEFT/CSULB collaboration, which is administered by CSULB-University College and Extension Services (UCES).

SEFT/CSULB is a partnership between member companies and the university. Each member company has representatives on the executive board, which is the policy-making body, and the technical committee, which develops curriculum and assists the board. Through the technical committee, industry has direct input into course topics and curriculum. The SEFT Program Director, a member of the CSULB-UCES staff, works with the executive board and technical committee to facilitate the development of SEFT activities and manages day-to-day operations.[1]

All four collaborations have well-defined communication structures to aid program administration, which are as follows:

- Florida Atlantic—Joint Advisory Board
- SEFT/CSULB—Executive Board and Technical Committee
- Texas Tech—Industry Advisory Board
- American University/CSM—Department Chair (American University) and CEO (CSM)

Funding arrangements reflect the diversity of missions and organizations involved. Florida Atlantic relies upon industry members and some state funding, while American University garners money from client fees. Texas Tech obtains grants and contracts from its corporate partner, while SEFT/CSULB operates on activity fees from classes and annual membership fees.

Successful Programming

Table 3 identifies the software engineering education and training programming of the four collaborations.

The four collaborations share common strategies for successful programming. Industry perspectives are heavily emphasized.

Each collaboration bases its programming on an assessment of the industry customer's needs. For example, in the SEFT/CSULB program, courses are often customized vs. being built from the ground up, as the industry customer's perspective is to develop and provide timely, short, and relevant materials and programs. All stress flexibility in program formats to adapt to their industry customers' schedules. For instance, each three-credit American University/CSM course is presented in a compressed format (i.e., five full days of instruction in one week or over three weekends). All quickly adapt their programs based on customer feedback.

Texas Tech approaches its software engineering education and training program for industry practitioners from a systems engineering perspective. The Raytheon Co. designates approximately 12 students per year to begin the 36-credit program, which takes three to four years to complete. The students come to Texas Tech for four weeks each summer for two summers, completing nine credits each summer. The program's other 18 hours are achieved in various ways, by taking courses at Texas Tech or elsewhere via distance education during the fall and spring semesters. The emphasis is on industry input and immediate adaptation, if needed.

American University's offering, like Texas Tech's, is a concentrated format of courses taken over a one- to two-year period on weekends; however, its core topical emphasis is a combination of systems engineering and project management. SEFT/CSULB differs; its curriculum is built around one- to three-day seminars. American University's and Florida Atlantic's offerings add an additional clientele in that their courses are available on both a for-credit (creditable towards a master's degree), and noncredit (certificate of completion) basis.

Several survey respondents noted the importance of appropriate instructor selection for collaboration programming. American University's program emphasizes the instructor's credentials, including both academic and industry experience/credentials, and approval by both the university and industry participants in the collaboration. SEFT/CSULB shares this perspective, as industry experience is critical for instructor credibility with practitioners.

All four collaborations perform evaluation activities. Customer satisfaction and revenues are common evaluation categories. In addition, Florida Atlantic and SEFT/CSULB appraise improved professional performance through post-program evaluations. Florida Atlantic collects metrics on the number of students enrolled, the number of students continuing, and the number of students applying for Florida Atlantic master's programs. SEFT/CSULB monitors quantitative metrics from their needs assessment and four levels of training program evaluations based on the Kirkpatrick Model.[4] Texas Tech's evaluation process is informal and relies on participant feedback to individual instructors.

Lessons Learned

While industry and academia are different in many ways, the surveyed collaborations have found ways for industry and academic partners to work together successfully for mutual benefit.

Table 4 documents some of the lessons learned shared by all survey participants.

Table 4. *Lessons Learned*

Preliminary Meetings
Define goals and needs clearly and candidly.
Identify decision-making structures within respective organizations.
Gain support of high-profile leaders from all involved organizations.
Win commitment to open communication, mutual support, and trust from all involved organizations.
Consider collaboration among competitors; it can be productive if approached correctly.
Assess organizations' "fit" before committing.
Commitment
Formally document collaboration agreement.
Select a governance structure to match joint goals and resources.
Dedicate resources.
Develop and document collaboration processes.
Develop communications structure (e.g., Industry Advisory Board, Technical Committee).
Operations
Adopt a customer-service orientation.
Be proactive.
Develop metrics to evaluate program success.
Consistently evaluate delivery formats and curriculums from industry perspective.
Seek feedback often and adapt programs immediately.
Focus on delivery time lines (especially university partners).
Incorporate "real-life" experiences and practical applications into activities.
Be selective of instructional staff. Look for practical industry experience, as well as academic credentials.
Facilitate frequent interactions among partners' staffs.
Emphasize mutual benefits as often as possible.
Assist partners whenever possible.
Growth
Participate in professional networks and organizations to identify potential new collaboration partners and customers.
Review existing contacts for potential new collaboration partners.

Conclusions

There are significant benefits derived from the interaction of universities and industry to meet the professional development, education, and training requirements of software engineers. All of the collaborations in this study cited increased university/industry appreciation and awareness, which led to better relations and mutual trust. These partnerships also resulted in increased potential revenues among the partners and an expansion of contacts and resources from both sides of the partnerships. Additional benefits included enhanced reputations for all involved and increased business development opportunities with other similar and affiliated type programs. For university faculty, the collaborations provided much-needed exposure to practical applications and industry trends and, in turn, enhanced faculty development, equipment procurement, and other resources for research and consulting. For industry students, the collaborations offered relevant courses in locations, times, and delivery formats that were realistic in the face of their workplace demands.

What makes a collaboration successful? Practitioners identified shared goals, planning, mutual trust, effective communication, and large doses of patience and hard work as necessary ingredients. As one survey respondent put it, "Software engineering education and training issues require a hybrid type of organization to address and act upon the elements of the situation in a timely fashion. Collaborations offer a structure/dialogue/action format for addressing the dynamic education and training needs of software engineers."

Acknowledgements

The authors would like to thank Nancy Mead, of the Software Engineering Institute, for continued Working Group facilitation.We would also like to thank the following individuals for participating in follow-up interviews: Bill Flury, American University/Center for Systems Management; Neal Coulter, University of North Florida; Michael Lutz, Rochester Institute of Technology; Cynthia Parish and Wendy Lasher, Software Engineering Forum for Training, California State University/Long Beach (SEFT/CSULB); and Don Bagert, Texas Tech University.Additional thanks go to Gregory Cohen of The Boeing Co. for technical editing assistance.

References

1. Beckman, K. (1997), *Directory of Industry/University Collaborations with a Focus on Software Engineering Education and Training, version 6*, SEI Special Report CMU/SEI-97-SR-018.
2. Coulter, N. & Dammann, J. (1994), Current practices, culture changes, and software engineering education, *Computer Science Education* 5(2), pp. 211-227.
3. Beckman, K., Coulter, N., Khajenoori, S., & Mead, N., 1997 Industry/University Collaborations: Closing the Gap Between Industry and Academia. *IEEE Software*, pp. 49-57.
4. Kirkpatrick, D. L. (1994), *Evaluating Training Programs: The Four Levels*, Berrett-Koehler Publishers Inc.

Note

1. SEFT recently changed its name to Systems & Software Engineering Forum for Training (SSEFT). The name change resulted from an executive board's request to change the collaboration's mission, reflecting changing needs of members.

As an outgrowth of the National Software Council meeting in October 1994, the SEI invited a volunteer group of professionals from industry and academia to Pittsburgh in June 1995 to form the SEI Working Group on Software Engineering Education and Training. Nancy Mead, SEI, facilitates the group. The Working Group's mission is to improve the state of software engineering education and training practice in professional development by investigating issues, proposing solutions, and publishing state-of-the-practice information. If you are interested in joining the Working Group, contact Nancy Mead at nrm@sei.cmu.edu.

In 1996, members of the Working Group began assisting the SEI in its publication of the *Directory of Industry/University Collaborations with a Focus on Software Engineering Education and Training*. In 1997, the Working Group defined a model of collaborations formed by university and industry to address the education and training needs of software engineering professionals. In 1997-98, the Working Group administered a survey to the 23 collaborations in the United States, Canada, and Australia documented in the *SEI Directory*, version 6.

About the Authors

This paper is a joint effort of the Software Engineering Institute (SEI) Working Group on Software Engineering Education and Training [see sidebar]. Contributors include: Kathy Beckman, Abacus Technology Corp.; Jimmy Lawrence, Auburn University; Nancy Mead, SEI; George O'Mary, The Boeing Co.; Cynthia Parish, California State University, Long Beach; Perla Unpingco, Lockheed Martin Co.; and Hope Walker, Lockheed Martin Co.

Dr. Nancy Mead is senior member of the technical staff in the Networked Survivable Systems Program of the Software Engineering Institute, and a faculty member in software engineering, Carnegie Mellon University. She is involved in the study of survivable systems requirements and architectures, and the development of professional infrastructure for software engineers. Prior to joining the SEI, Mead was a senior technical staff member at IBM Federal Systems, where she spent most of her career in development and management of large real-time systems. She also worked in IBM's software engineering technology area, and managed IBM Federal Systems' software engineering education department.

Software Engineering Institute
4500 Fifth Avenue
Pittsburgh, PA 15213

Perla Unpingco is a Subcontracts Program Manager for major subcontractors in Satellite Control Systems. She has oversight responsibility for subcontractor performance, including technical, financial, and schedule adherence to directives and is a member of INCOSE.

Kathy Beckman, a Process Improvement Engineer with the Software Productivity Consortium in Herndon, Va., is an ASQ Certified Software Quality Engineer and author of *Directory of Industry/University Collaborations with a Focus on Software Engineering Education and Training*, an SEI Special Report.

Hope Walker is a Program Management Consultant and instructor at Lockheed Martin Corp. She has 18 years of experience in software development, systems analysis, and management on Air Force and FAA contracts.

Cynthia L. Parish has served as coordinator of adult learning for University College and Extension Services at California State University, Long Beach since 1996, and has been with CSULB-UCES since 1993. Parish has also served as director of Systems and Software Engineering Forum for Training since 1998. Previously, Parish worked for the L.A. Times, where she was instructional development project manager for the Integrated Circulation Information System.

George O'Mary is a senior software process specialist in Software Process Improvement at The Boeing Co. in Huntington Beach, Calif. He has a bachelor's degree in accounting from Bellarmine College in Louisville, Ky. He is in the core Software Engineering Process Group and has more than 20 years experience in the design and presentation of technical training. O'Mary has successfully completed the ISO 9000 TickIT Auditors' Training Course and Team Training for the Software Engineering Institute's Capability Maturity Model®-Based Appraisal for Internal Process Improvement. He is also the Boeing technical representative at the Systems and Software Engineering Forum for Training.

ACPINS Makes Management Easier

Are you scratching your head trying to identify the most current version or revision of your software? Do you know who is using it and where? Do you need customized management reports for all those briefings? The Automated Computer Program Identification Number System (ACPINS) can help.

Air Force Materiel Command (AFMC) projects a 14 percent growth in software inventory by fiscal 2000. If all of the Air Force sees this kind of expansion, configuration management will become even more challenging. The ACPINS is a tool that can make this job easier for developers, users, and managers.

Managed by OC-ALC/TILUC, CPIN System Section, at Tinker Air Force Base, Oklahoma, this online standardized Automated Information System collects and maintains data used to identify, manage, catalog, requisition and distribute Embedded Computer Resources software for the Air Force. ACPINS supports the global software needs of defensive weapon systems, tactical systems, aircraft, missiles, ships, communications, command and control, and spacecraft.

AFMC supports approximately 5,300 embedded computer systems. This also includes approximately 98,466 associated Computer Program Identification Numbers (CPINs) assigned to computer software configuration items and their related engineering documentation packages. Projections for fiscal 2000 indicate AFMC will support 9,225 embedded computer resource systems, and approximately 115,000 related CPINs.

Unique Designators Help Tracking

Computer software configuration items and related engineering documentation are developed simultaneously along with hardware and support equipment when a weapons system is acquired. These are identified by computer program identification numbers—standardized, unique designators used to track the configuration of computer software configuration items and related engineering documentation during its life cycle. The CPIN identifies product baseline software, coexistent versions, and revisions that occur after the baselined item or version is distributed.

CPINs often are requested and assigned during the full-scale development phase, prior to the critical design review. However, for systems or subsystems that are past the critical design review, the request is made as soon as possible afterward. Early assignment of a CPIN enables the software developer or manager to include:

- The CPIN identifier in the documentation.
- The contents of the operator instruction manuals, or applicable technical orders.
- An identifier that can be affixed to the program media (tape leader, disk pack, etc.).

A CPIN identifier is also used on the title page of the engineering documentation package.

Assigning a CPIN early in the life cycle of computer software configuration items also allows indexing in the CPIN compendium.

CPIN compendiums are consolidated indexes, which list CPIN identifiers and related information. The compendiums announce pending computer software configuration item releases, reflect status of computer software configuration items, and provide timely information and descriptive data on new, updated, current, and inactive software and related engineering documentation. The CPIN compendiums also are used to identify software needed for research purposes, to update files and records, to reference inventory, and to establish requirements for initial distribution and one-time software requisitions.

All compendiums are available online as well as in microfiche form. The forward of each microfiche compendium contains general information relative to the CPIN System and detailed instructions for using the compendium. Microfiche compendiums are produced as funds are available.

There are five general types of CPIN compendiums: index of compendiums, cross-references, Air Force compendiums, command compendiums, and country compendiums. They are briefly described as follows:

- Index of compendiums provides managers and customers a current list of CPIN compendiums and cross-references.
- Cross-references are designed to serve as quick references or research aids for CPIN association to selected data elements.
- Air Force compendiums contain lists of CPINs and related information. They are updated through daily processing in the ACPIN system as revisions, and are available online, or are published in microfiche form no more frequently than every 180 days.
- Command compendiums list only command-managed CPINs and related engineering documentation.
- Country compendiums contain lists of CPINs and cross-reference data, which are applicable to a specific foreign country.

System Boosts Mission Capability

In addition to enhancing configuration management, ACPINS offers the Air Force increased mission capability, convenience, customized management reports, and security.

ACPINS boosts mission capability by allowing customers/users to almost instantly see information about newly assigned basic CPINs, revisions, versions, updates, and changes. Online compendium changes are up-to-the-minute, and give technical order distribution offices (TODOs) a heads-up to review their requirements. Computer software requirements lists are available through e-mail in minutes, instead of days or weeks. In addition, TODOs can have their requirement request—Order (AFTO 157)—approved and their software shipped from the Software Control Centers (SCCs) in just hours.

The system also eliminates duplication, therefore eliminating excess production costs, for software centers and managers. Questions about weapon system order issues are resolved as they surface. Mission capability will increase even more with additional system improvements on the horizon. The online ACPIN System has taken advantage of technology and transitioned to a Web-based system.

ACPINS Automates FMS Approvals Process

One of the conveniences of ACPINS is the faster handling of requests for Foreign Military Sales/Security Assistance software and compendiums. Customers submit an Order (AFTO Form 157) request, which contain an Air Force TODO code assigned by a Security Assistance Technical Order Distribution System (SATODS). The request is forwarded through a country TODO, the CPIN System Section Foreign Military Sales (FMS) point of contact, or to the prime managing center.

The data is entered into the database, where ACPINS verifies case status. If the case is current and reflects a monetary balance sufficient to pay for the items, the request is processed. When shipment is completed, shipping information is entered into the database and transmitted by daily interface to the SATODS.

FMS compendiums and cross-references may be accessed by Software Control Centers, equipment specialists, and program managers. Approvals/disapprovals by these individuals for country requests are processed online. Specific access will be available for the Foreign Disclosure Officers.

At this time, foreign nationals obtain compendiums and cross-references on diskettes or microfiche. Future plans include producing compendiums on compact discs. Later, FMS customers will have access to the Web.

ACPINS Customizes Management Reports

Collected and stored data for each software item, and related engineering documentation, may be extracted from the ACPIN database and formatted into various customized management products. They assist software managers at all levels in accomplishing configuration management and provide managers an overview of software systems, subsystems, related applications and documentation packages. These products are available online and may also be obtained by requesting the report from the CPIN System Section.

Reject notices are produced by the ACPINS database as transaction process, or Software Control Centers and the CPIN System Section may produce notices for mailing. Mailing and media identification labels also are produced by the Software Control Centers and the CPIN System Section as needed.

System Processes Unclassified Data

All data processed within the ACPIN System is unclassified. Data elements may relate to classified software and/or engineering documentation packages, but no classified information is entered in, processed, stored, or output by the ACPIN System. Access to the system and the databases is managed through system controls and customer passwords based on multilevel access approvals granted by the ACPIN System Managers.

Firewalls also are installed and encryption is in place on the Web-based system. Most ACPINS customers already have user identification and passwords for system access.

ACPINS database access is provided via personal computers through a communications network using compatible hardware. Access is available to Air Force software developers, system managers, system program offices, inventory managers, and equipment specialists, software managers and engineers, SCCs, Major Commands, Development Engineering Prototype Sites, Air Force Meteorology and Calibration Program, and other users.

ACPINS Traces Roots to Batch Processing

The original CPIN System began as a manual system using single manager storing data in a technical order configuration consisting of check-tapes. The system integrated to batch process and evolved into today's automated online system. The concept of a distributed database to facilitate tracking the software began in 1989. However, as the needs of the customers grew, and the system took shape, the concept changed to a centralized distributed processing database with network access.

The CPIN System today processes data on the E3000, Sun Sparc20 Server in a UNIX environment. Data is entered through personal computers or SUN workstations, which interface with the central database located in the OC-ALC CPIN System Section. Information includes Numbering (AF Form 1243), ACPIN Data and Control Record, data which establishes and maintains CPIN records, and Orders (AFTO Form 157), computer program configuration item request, data which establishes and controls software requirements and distribution. Existing data systems satisfy storage of selective portions of software configuration management data and customer requirements, which are output on paper, microfiche, and online products.

The HQ Air Force Materiel Command, office of primary responsibility, carries out overall management duties and provides general policy and guidance for the ACPIN System. Computer Resources Support Improvement Program (CRSIP OO-ALC/TI-3) is acting as the Configuration Control Board for the modifications of the ACPINS.

OC-ALC/TILUC serves as the mission activity responsible for the ACPINS operation, budgeting and funding for maintenance, and life cycle management.

Increased demands on managers' time and resources has made it even more important to use existing assets to the fullest extent possible. Utilizing the ACPIN system is the most efficient and cost-effective way to do business.

About the Author

Gerald Ozment is Chief of the CPIN System Section at Oklahoma City Air Logistics Center. He is responsible for enhancing and managing the ACPINS. In addition, the section is the Air Force and AFMC focal point for the development, identification, indexing management, and establishment of customer requirements for embedded computer software used in most weapon systems and support equipment in the Air Force. Ozment has been associated with the system for approximately 18 months. He has spent 40 years in federal service, 21 years of which were in the Air Force.

OC-ALC/TILUC
7851 Arnold St, Ste 205
Tinker Air Force Base, Okla.73145-9147
Voice: 405-736-2227 DSN: 336-2227
Toll Free: 888-742-4461
FAX: 405-736-7734 DSN: 336-7734
E-mail: Gerald.Ozment@tinker.af.mil

A Concept of Operations for Product Lines

Software product line technology has been adopted to achieve savings while improving product quality and delivery time. To achieve these results, several Department of Defense organizations developed a process for fielding the product line and captured that process in a concept of operations (CONOPS). This article introduces the concepts behind the CONOPS and provides guidance for an organization wishing to adopt a product line approach.

Moving to a Product Line Approach

Your organization has studied the advantages of product lines [1, 2] and is developing an acquistion approach [3]. What comes next? When an organization decides to move to a product line approach for acquiring or developing software, it must address several key issues:

1. What constitutes the product line?
2. How is the product line introduced?
3. What are the key organizational elements involved in defining, developing, and fielding the product line?

The operational concept for a product line should be documented as a CONOPS. *CONOPS* conveys the operational nature of the process for fielding a product line. An organization develops a CONOPS to establish the desired product line approach it wishes to take, and to document decisions that define actions and organizational structure needed to put the approach into operation.

Aspects of CONOPS Peculiar to Product Lines

The concept of operations for a product line will contain:

1. The strategies, tactics, policies, and constraints that describe *how* the process will be used to field the product line.
2. The organizations, activities, and interactions that describe *who* will participate in fielding the product line and *what* these stakeholders do in that process.
3. The specific operational processes, in overview fashion, that provide a process model for fielding the product line in terms of *when* and in *what* order these operation processes take place, including dependencies, concurrencies, etc.

Application of CONOPS to Core Asset, Product Development/Acquisition

A CONOPS should address a number of key product line issues, both for core asset and product development. An organization needs to address these issues as it makes product line decisions. For product development, the CONOPS helps address needs of program managers, developers, and others in product oversight or decision-making roles. Issues may be grouped into categories as shown in Table 1.

An organization develops a CONOPS to establish the desired product line approach it wishes to take. The CONOPS should contain a detailed description of this approach, including decisions defining the approach and organizational structure needed to put it into operation. The CONOPS may possibly present alternatives.

Specific CONOPS Practices

The CONOPS may not be strictly

Categories	Core Asset Development	Product Development
Key decisions	Process and organization for developing core assets; key action steps for putting the CONOPS into effect	Process and organization for developing products in the product line
Components	Known components or elements in the product line including the product line scope, the architecture and other assets, and the product line activities	Effects of using product line assets in developing products
Context	Relationships among the stakeholders and sources for asset development: legacy systems and assets, asset developers and product users	Relationships among the stakeholders and assets for product development: product line assets, asset developers, product developers and product users
Activities	Sequence of activities moving from product line scoping, through architecture, and component development. Product line sustainment	Activities for using core assets in the development of individual products
Organizational elements	Organizational elements and the role they play in fielding the product line	Organizational elements and the role they play in the development of product line products
Rationale	Rationale for moving to a product line approach as well as risks	Rationale for using product line assets as bases for product development
Integration	Tie together the above elements to provide guidance in development activities such as the development of component assets and the use of the architecture and assets in producing products	Production plan for products in the product line. Guidance is especially important for reflecting the results of using core assets in product developments to support their continued improvement

Table 1. *Key product line issues to be addressed by CONOPS.*

followed due to costs, scheduling, performance, capability, or insufficient commonality. For example, a set of requirements for a new system may fall outside the bounds of the existing assets. The product line organization must determine if:

1. This system should be developed outside the product line.
2. The new requirements should become an area for continued growth within the product line.
3. Potential users should be encouraged to tailor requirements to capabilities already accomodated within the product line.
4. Establishing a new product line is recommended.

All of these factors must be considered as part of the business analysis for meeting the needs of candidate users and should be spelled out in the concept of operations or accompanying documents.

Understanding potential user needs, implementing solutions, and managing product evolution goes beyond creating an architecture and components. It requires a systematic and comprehensive approach (i.e., the product line concept of operations) to marshal existing resources and identify additional methods of lowering costs of providing thorough use of product line assets. Strong management support is key, and identifying a champion who will assume responsibility for managing and facilitating the effort.

The following practices are essential elements to be considered in putting the CONOPS into effect:

Selecting a Product Line Approach Champion

The champion must be the owner of the CONOPS and employ available resources in concert with each other according to the plan. The champion is responsible for defining and articulating the integrated vision for assets and for the product line CONOPS. Providing for the existence of an architecture is not enough to ensure a product line will result. It takes a concerted, well-coordinated effort to overcome technical, cultural, political, and programmatic obstacles.

Architecture-Based Development

CONOPS results are predicated on the use of architecture-based development. Much is implied by this approach to system design. Fundamental is establishing a development process centered on a software architecture to address common and mission-unique requirements and applied to developing the system in a prescriptive manner.

The CONOPS should help an organization integrate its architecture-based development plans to manage, design, implement, and test product line assets as well as systems in the product line. The CONOPS describes elements of architecture-based development:

- A set of program plans (program management, systems engineering management, software development, configuration management, test and evaluation, integration, etc.).
- The architecture description document.
- A set of architectural templates or tools that automate the representation and use of architectural templates.
- Typical development tools including those for detailed design and coding, configuration management, compilers, graphical user interface builders, etc.
- Documentation tools.

Program plans identified by the concept of operations should establish the management infrastructure and reporting elements similar to the structure of the architecture. The CONOPS should describe the estimation and tracking processes that are keyed to this structure. The CONOPS should also indicate how prototyping, evaluation, or other efforts will support validation of architectural decisions.

Planning the Impact of Transition to the CONOPS.

The transition to a product line strategy may require significant change in existing organizations. Any plan for transition must address the impact of change to organization, management, and acquisition elements.

Organization

The product line approach requires special attention to bring together core competencies from across existing organizational structures. The concept of operations spells out organizational restructuring that will enable concentration and sharing of personnel and skills, leading to greater overall productivity.

Management

The concept of operations must address management strategies that support the product line. New incentives will be needed to support the management and use of a product line approach. The CONOPS must address managerial changes that come with adopting a product line approach, including:

- Establishing promotion and reward structures.
- Breaking the *not invented here syndrome.*
- Integrating efforts across organizational boundaries by relying on support and assets from other parts of the organization or other organizations.
- Recognizing that the development of mission-unique applications requires more than just component integration.

Acquisition

The CONOPS must address investment strategies that support a series of systems based on a common infrastructure. A CONOPS makes the case that systems need to be acquired through methods that encourage using the existing product line infrastructure and leveraging existing assets. Acquisition must also include direct support for sustaining and enhancing the infrastructure to support future needs.

Alternative approaches should be considered in formulating financial aspects of the CONOPS. These may include:

- Pooling funds from all the systems that fall within a product line to pursue product line development.
- Designating a single program to manage the common infrastructure.
- Using existing commercial-off-the-shelf software products.

Product Line Asset Utilization

A concept of operations establishes procedures to ensure every proposed program is examined for similarities with existing systems in mission and underlying functions. The goal is to focus new development on unprecedented areas and reuse product line assets as much as possible. Reuse of assets includes much more than software components. Design, architecture, requirements, and models are all assets for reuse. The acquisition approach for accommodating new programs should encourage leveraging past investments to the fullest and contributing assets for use in future efforts.

Support Strategy

A basic element of a concept of operations strategy is continued maintenance and enhancement of product lines and corresponding architectures. A CONOPS defines the organizational structure that provides this support and its interaction with product development. Updated assets are provided to various customers/users according to maintenance/upgrade agreements established at the initiation of development of new members of the product line. Maintenance and support of the product line architectures and components are a natural consequence of the product line development strategy.

CONOPS Risks

Failure to Identify a Product Line Champion

Success of the product line requires strong management in the form of a product line champion. For most organizations, the nontechnical challenges alone will limit success unless one individual is given and assumes management responsibility. Technical activities involved in fielding a product line, from conceptualization to asset development to producing the first products may take two or more years. The champion must maintain the vision during this *black hole* period. In particular, the product line champion needs to take early initiative and oversee development of a concept of operations to solidify the conceptual approach and obtain the buy-in of key stakeholders.

Lack of Appropriate Product Line Vision

A CONOPS will often be written a year or more before assets are built and products start to flow from the product line. Developers must focus attention on where the product line should be three to 10 years hence in order to plan for full transition. The organization must be able to address the development of assets, their use and refinement in specific products, and potentially, transition of the product line approach throughout the enterprise.

Failure to Maintain the CONOPS.

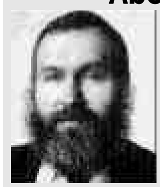
The concept of operations is not meant to be completed and placed on a shelf. It should be constantly reviewed and revised as the product line is fielded and the product line evolves. As a document released early in the process of fielding a product line, the CONOPS can only provide a starting point for product line development. Lessons learned in asset development, initial product development using assets, and sustainment of the assets must be factored back. If the CONOPS is not maintained in spirit, if not as a formal document, the product line may not successfully evolve to address new customer needs.

To help offset these risks, *Guidelines for Developing a Product Line Concept of Operations* [5] has been developed that can be suitably applied by an organization to meet its specific needs and circumstances. The guidelines and scenarios help an organization that has proposed a product line. This document provides exerpts from sample CONOPS and details the class of information to be contained in each section of a CONOPS.

References

1. Clements, Paul. *Software Product Lines: A New Paradigm for a New Century.* CROSSTALK, February, 1999, pp. 20-22.
2. Clements, Paul; Northrop, Linda M., et al. *A Framework for Software Product Line Practice.* Pittsburgh, Pa.: Software Engineering Institute, 1999. Carnegie Mellon University. Available at http://www.sei.cmu.edu/plp/framework.html
3. Jones, Lawrence. *Product Line Acquisition in the DoD: The Promises, The Challenges* CROSSTALK, August, 1999. pp. 17-21.
4. ANSI/AIAA. *Guide for the Preparation of Operational Concept Documents* ANSI/AIAA G-043-1992
5. Cohen, Sholom *Guidelines for Developing a Product Line Concept of Operations* (CMU/SEI-99-TR-008). Pittsburgh, Pa. Software Engineering Institute, 1999.

About the Author



Sholom Cohen is a senior member of the technical staff of the Software Engineering Institute, where he has worked for more than 12 years. As member of the Product Line Systems Program, Cohen has authored major technical reports and conference papers on product line technology, domain analysis and domain engineering methods. He is the author of reports on Product Line Concept of Operations for the Air Force Electronic Systems Center, the National Reconnaissance Organization, and DoD test and training ranges. Prior to joining SEI, he was a member of the software engineering technology branch of the McDonnell Douglas Astronautics Co., where he was a key developer of Common Ada Missile Packages components and tools.

Sholom Cohen
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pa. 15217
Voice: 412-268-5872
Fax: 412-268-5758

Attention Authors–CROSSTALK Themes Have Changed

Due to an anticipatedly high level of submissions, our CMMI℠ theme will carry over to a second issue (Aug. 2000), for which the deadline is April 3. Other upcoming themes and submission deadlines as amended are:

September 2000	October 2000	November 2000	December 2000
COTS	*Network Security*	*Software Acquisition*	*Project Management*
Deadline: May 1	**Deadline: June 1**	**Deadline: July 1**	**Deadline: August 1**

Please submit potential articles to:

OO-ALC/TISE,
Attn: CROSSTALK Editorial Staff
7278 Fourth St.
Hill AFB, Utah 84056-5205

Fax: 801-777-8069 DSN 777-8069
Voice: 801-775-5555 DSN 777-5555
E-mail: crosstalk.staff@hill.af.mil

You may visit http://www.stsc.hill.af.mil/Crosstalk/crostalk.html to obtain updated guidelines for CROSSTALK authors.

Managing the Changing Mainframe Environment

Years ago the mainframe was proclaimed dead, to be replaced by client/server environments. However, 70 percent of organizations' data today still remains on these super-systems. The Internet has given way to a new set of technologies that enable organizations to leverage these legacy systems and their unmatched processing and storage power without re-engineering. But what are the short and long-term benefits of these solutions?

Nearly everyone resists change, largely because we are required to expend more effort than normal adapting to it. This is especially true in the mainframe user world. As the state-of-the-art changes, users have had to adapt to changes in menu selections, applications, even the equipment used on the desktop.

Most 3270, 5250, and VT200 terminals have disappeared. In their place, personal computers of various types provide much more capability, while offering the familiar mainframe application screens. DOS has vanished, replaced by Microsoft Windows, which permits simultaneous operations and cut-and-paste capability. Users have had to adapt to all these changes. Now client/server and Web-based applications are appearing, forcing another change upon users. As with all transitions, this one is easiest if it can be done without causing complete disruption and user re-education.

The Changing Mainframe Environment

Client/Server

Client/server applications promise an environment that uses the network as an enabler of processing power on the desktop. But the mainframe computer is not dead. In fact, it is enjoying a strong comeback as organizations realize that a server-based environment does not always address issues of scalability, reliability, and security. Client/server applications can require huge amounts of bandwidth. Local area network (LAN) technologies easily provide fat pipes to run applications, but organizations linked to one another or to portable or mobile users may not be so fortunate.

Wide area networks still lack a LAN's cheap bandwidth, especially to international locations. This can cause serious problems with client/server applications that may assume ethernet or higher speeds from server to workstation. This is not meant to defile client/server applications, but rather to illustrate the best way for an organization to offer service may continue to be a mixture of the traditional mainframe and client/server. A rush to throw the mainframe out for the sake of *modernization* may be counterproductive. If the conversion from mainframe basing to client/server is justified, the conversion process usually represents a significant modification of user behavior when interacting with the data.

Wider Need for Strategic Information

Internal change is stressful, but technological changes in the past few years have also brought an external element into the mix. Other organizations may now need to access information that has traditionally been unavailable, or available only as printed material. Printed material is outdated nearly as soon as it is printed, so direct access to information is more desirable. If several external organizations require access, there will be a need to provide transitional tools to them as well. Web-based access becomes extremely valuable in this case, as each external user may be presented with a custom interface to the information.

The Mainframe Answers Back

TCP/IP Networking

Access to mainframe computers has become easy in recent years. Where System Network Architecture or DECNet were once the only game in town, Transmission Control Protocol/Internet Protocol (TCP/IP) has taken over. The Department of Defense (DoD) development of the Internet provided a growth medium that has spawned an incredible commercial communications force. That has benefited the DoD with much wider access and higher bandwidth. Users can connect to an organization's mainframe computer from anywhere on earth (or in space) through a dial-in TCP/IP connection. Security concerns are still valid and more important than ever, but access is now an administrative, rather than a technological, issue.

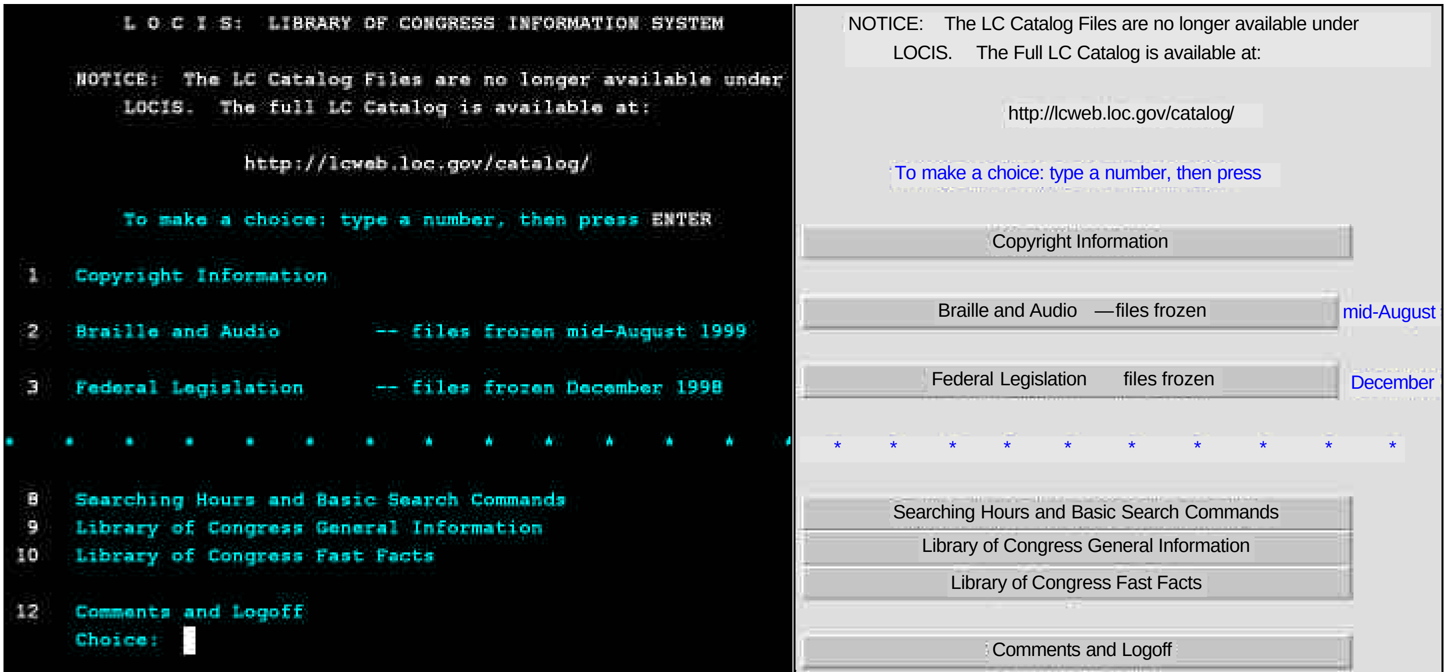
The explosive access to mainframe systems and changing mission requirements has made organizations focus externally. Information that was once only for internal use or occasionally available by request is now required to supplement real-time decisions. Tasks involving outside organizations can be accomplished more quickly with immediate access. A mechanism to offer the information online streamlines the entire workflow process. Yet, as data access opens up, there are fewer people available to install and maintain the systems to do the job. Many talented people leave government service for higher salaries in the private sector. Those who remain have a challenge to provide more and better services with less budgetary support.

Alternative Strategies

From Gooey to GUI

One of the most cited reasons for going to client/server technology is the provision for a graphical user interface (GUI), which presents a more orderly and intuitive entry method than traditional mainframe menuing systems. However, for users who have been entering mainframe data for years, this change can be intimidating. What if there were processes to ease users into the GUI?

As it happens, there are such processes. Either as an interim step, or as a way of postponing a large client/server conversion, transformation of mainframe application "green screens" into Web-style interface can be accomplished. Green screen rejuvenation uses existing application screens, and overlays Windows button and mouse techniques to provide an easier navigation path. If a GUI is the primary requirement for going to client/server architecture, rejuvenation should be considered as a viable alternative to a complete system change. With out-of-the-box solutions such as J42, provided by J&B Computing Services and integrated into



A typical mainframe application green screen before and after J42 rejuvenation BEFORE/AFTER

WRQ Reflection EnterView (www.wrq.com/enterview), this transformation can be done automatically. As a result, the mainframe system may be completely disguised from the user and presented entirely as HTML pages on a web site.

The Browser—A Unified GUI Standard?

The explosion of the World Wide Web has made nearly everyone who touches a computer familiar with Web browsers. New technologies are bringing Web-based terminal emulation to the fore, providing a means to have a single user interface accessing applications that may be based on a Web server, then browsing to data on a mainframe with a traditional terminal interface contained on a Web page. Many client/server applications are offering a browser interface as an alternative form of presentation, which makes the browser an ideal transition display for changing systems.

As an added benefit, HTML is relatively easy to write, change, and maintain. System look and feel can be modified to suit individual groups, or in extreme cases, individual users. The addition of links to written or graphic help files are also easily included, which makes Web technology an ideal vehicle to present training along with the new application look. As a bonus, the help files can be made interactive with JavaScript, allowing users to be guided directly to the proper sequence of actions.

Security always heads the list of concerns when external access is discussed. Web-based host access models are available now with encryption through proxy servers that can provide 168-bit DES protection and close control of user access. This technique can be applied to internal networks as well for protection of sensitive information within the organization.

Alternate Strategy

An organization might wish to modernize an application with client/server technology and find that the expense is prohibitive. However, the benefits of a simplified interface to users are still desirable. In these cases, use of rejuvenation tools and browser-based clients will simplify and streamline tasks residing on the mainframe. HTML also allows the presentation of mainframe data in more convenient or effective formats. Graphic charts may be included for tabular data without custom mainframe programming or applications. Special fonts such as large typefaces may be included for emphasis or use by the visually impaired. Changes can be quickly and easily made if the format is ineffective or if emphasis changes. Many organizations now have dedicated HTML writers who are responsible for external Web site content and appearance, who can also be helpful in interface construction.

Another benefit is that the mainframe application code can remain unchanged. Legacy applications can be hard to modify and customize. Web-based access provides an opportunity to precisely define presentation styles and content without having any program modification. The ability to have multiple sessions active and provide information to the same page can enable viewing of data relationships in a way that has not been available before. Even in the client/server model, this view can be difficult or impossible to provide if the information is present within the organization but not part of the server database.

The browser makes an ideal access point for the information. Most personal computers are equipped with either Netscape or Internet Explorer. Traditional full-featured host access clients often occupy tens of megabytes of storage space. While this is usually not of concern in laptop computers with multigigabyte hard drives, using browser-based host access frees up the majority of the space by allowing the client to be downloaded only when necessary. If network speed is a concern, host access based on Java technology can be persisted in the browser on the first download. Applets stored in this fashion take up much less disk space than traditional clients. Version management becomes extremely simple. When a persisted applet is run, the version number is checked, and if a newer version of the applet is available on the

Web server, it is replaced. Desktops need never be touched, and all changes to applet versions can be done once at the server level. Similarly, if access to new systems is required, or withdrawn from old systems, a single modified configuration parameter will suffice for all users concerned with the change. Other options include providing both a standard green screen interface and a rejuvenated interface using the same client applet. A number of different implementations can be made and offered to different groups of users. It is even possible to have several host sessions presented on a single HTML page, or to launch multiple separate windowed instances from a single HTML link.

Rejuvenation cannot answer the issues of operating expense and support staff required to maintain a mainframe computer. But if the drivers of change are primarily access to mainframe information and appearance, Web-based host screen rejuvenation becomes a competitive alternative to client/server conversion.

The Human Factor

Change in the mainframe host environment is never trivial and often overwhelming. With the right tools, it can be accomplished with less pain than expected. Web-based host screen rejuvenation is an example of an alternative solution to client/ server requirements. Conversions to Web-based host access are themselves change agents. In the traditional mainframe world, there was not much need for interaction with computer resources outside the information technology department. With the advent of personal computers, many organizations have formed groups that specialize in desktop, LAN, and Web deployment. These groups have a different reporting structure than the mainframe support department. It can be a challenge for such potential competitors to work together to provide support for a common user group. When another organization is involved as a user or provider of some data, cross-organizational differences can completely overshadow technical aspects of any project. Users have become more active as they are asked to take on additional responsibilities with the shrinking and consolidation of organizations. Their demands for ease of use, often traditionally ignored, must now be considered carefully in any system design. The downfall of the mainframe and complete switch of mission critical applications to client/server environments has been predicted since the first commercially available personal computer appeared in the early 1980s. This has not been the case, and access to mainframe-based applications is experiencing new interest as Web technologies widen their appeal. Client/server environments still offer advantages in some situations, but the wise information technology manager will realize that there are few instances where one size fits all. Do not be afraid to look at interim solutions or other alternatives to disrupting your users. After all, are *they* not the reason we all work so hard in this business?

About the Author

Doug Wetzel is a systems engineer, specializing in the network management needs of WRQ's government customers. His technical engineering background also includes work at American Satellite Co., AT&T, Hercules Aerospace Corp., and US WEST. He enjoys digital and space communications, amateur radio technology, and flying airplanes. WRQ is a provider of software that enables information technology to connect, manage and secure networked environments, including WRQ Reflection host access software, which connects PCs with IBM mainframe, AS/400, UNIX, Digital and HP host systems.

Doug Wetzel, Government Systems Engineer
WRQ Inc.
1500 Dexter Avenue North
Seattle, Wash. 98109
Voice: 206-217-7862
Fax: 206-301-6904
E-mail: Dougwe@wrq.com

CMMI^SM Draft Models Released and Reviewed

PITTSBURGH—Public review of version 0.2 of an integrated model for systems and software engineering improvement, CMMI-SE/SW v. 0.2, was completed in November.

The Capability Maturity Model® Integration [CMMI^SM] Project developed the model. The project was a collaboration sponsored by the Office of the Secretary of Defense and the National Defense Industrial Association with participation by government, industry, and the Software Engineering Institute. The objective is to develop an integrated set of models, an assessment method, and training materials that provide support for process and product improvement in industry and government.

The project received 2,345 change requests during the public review, which began August 31 and included participation from 67 organizations and 105 individuals. Project members are gratified by the quantity and substance of the responses to the call for review and will process all comments to improve the content and form of the model.

Draft components of the integrated model that include support for integrated product and process development (IPPD) also underwent a public review period, which ended Feb. 15. With the IPPD model extension, organizations using different models for improving systems engineering, software engineering, and integrated product and process development will be able to use the integrated model to coordinate efforts to improve in all three disciplines. Using this CMMI^SM model encourages enterprise-wide improvement and integrated assessment of all three functions.

The integrated model incorporates the best features of the source models from which it is derived—Capability Maturity Model for Software (SW-CMM®) v. 2.0 draft C, EIA/IS-731 Systems Engineering Capability Model(SECM), and Integrated Product Development Capability Maturity Model (IPD-CMM).

It will enable organizations to build on previous investments in improvement based on the SW-CMM, the SECM, or the IPD-CMM, and benefit from the standardization and commonality of the integrated model.

For more information about the CMMI Project and the public release of IPPD model components, see http://www.sei.cmu.edu/cmm/cmms/cmms.integration.html

Bill Pollak, Team Leader, Technical Communication Public Relations Coordinator
Software Engineering Institute, Carnegie Mellon U., Pittsburgh, Pa. 15213-3890
Voice: 412-268-5656
Fax:412-268-5758
E-mail: wp@sei.cmu.edu

Web-Based Software Process Improvement Repository

This paper describes the structure and contents of the Space and Naval Warfare (SPAWAR) Systems Center, San Diego's web-based Software Process Improvement (SPI) repository. The Software Engineering Process Office (SEPO) has provided its software engineering Process Asset Library (PAL) in the public domain in order to foster continuous software process improvement throughout the software engineering community.

The Organization Process Definition Key Process Area (KPA) of the Software Capability Maturity Model® requires organizations to provide a repository of software engineering processes. SEPO, a Software Engineering Process Group at the SPAWAR Systems Center, San Diego has provided its repository at http://sepo.spawar.navy.mil to meet this requirement.

Organizations embarking on the SPI journey often do not know how to begin. Other organizations are implementing SPI, but have missing pieces. Some organizations have implemented SPI, but lack information about what other organizations have done that could help them improve their existing processes. The needs of these organizations can be met by sharing process assets and knowledge among the software process improvement community. SEPO began an effort to accomplish this by providing its software process assets and other SPI information on the Web and in the public domain.

Background

More than three years ago, our organization embarked on a modest effort to provide our product and process assets in electronic form via the Web. The primary goal was to provide software engineering information that could be located and quickly downloaded by our software projects. Hence, the Web page is plain and simple. As we worked through the site design, we decided to use our page not only as a repository for our process assets, but as a repository for all our assets, with a few exceptions, and available in the public domain. Our reasoning was three-fold.

First, many contractors provide support to our organization's projects. Putting our page in the public domain enabled them to have easy access to our products and saved us the administrative difficulties of giving out passwords and user accounts to gain access. It was also hoped that making the repository easily accessible to our contractor community would encourage the use of our products, thereby benefitting our organization.

Secondly, we wanted to share our process assets with the software community at large, hoping it would provide us feedback on our products so we could improve them. We have had many site visitors and lots of appreciation for availability of our products, but few comments on artifact content.[1]

Our third goal was to give other organizations implementing SPI efforts an example of how one organization implemented it. They could then begin their own program, fill in missing pieces of an existing one, or improve current processes using information from the repository. This is our contribution to spreading SPI.

Web Page Layout

The home page is made up of a main page that contains a navigational table at the top and information about "What's New," "Hot Topics," and "Upcoming Events" at our organization (see Figure 1). The rest of the page is divided into three sections titled "SEPO Resources and Software Engineering Information," which contains information about software process improvement at our organization; "Other Sources of Software Engineering Information," with links to other software engineering sites; and "Other Links," which contains links to sites on topics that may be of interest to software engineering professionals.

The Process Repository

The majority of the software process improvement repository can be found by clicking on "SW Eng Processes and SW Docs" located in the navigation table at the top of the main page. Clicking on this will bring you to "Software Engineering Processes by KPA" at http://sepo.spawar.navy.mil/docs.html (see Figure 2). Process assets are divided by KPA levels. Each KPA contains our organization's policy for it, a process definition (if one exists), and other supporting information.

Other information includes sample plans, templates, and other documents from our organization and others. We also include internal processes we use in the day-to-day office operations. We do not have a complete set of process assets for every KPA, but we are working toward that goal, and have placed many products on the Web page knowing they are not perfect. Our philosophy is to provide preliminary and in-process products to our software projects, so they may benefit from the latest process information.

Training modules exist for many KPAs, but have been intentionally left off of the Web page as we think training encompasses more than just presentation material. To get the full value from the training materials, it is best to attend the classes. Detailed descriptions for many of our training courses are available at http://sepo.spawar.navy.mil/training.html.

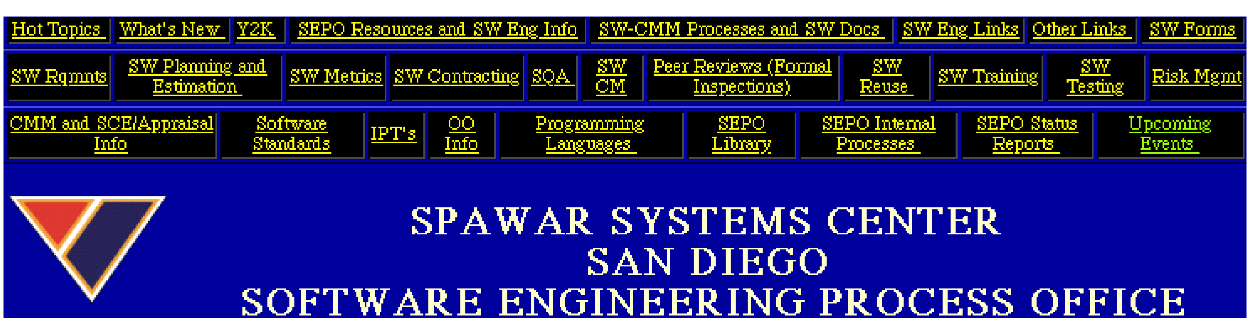


Figure 1. *The top of the SEPO web page includes a navigation table at the top of the page for quick access.*

Other Benefits of a Web-Based Repository

We met our original goals in development of the Web page, and there have been other benefits. For example, we no longer have to provide students with large volumes of course reference materials for our training courses because they are now available electronically. In one instance we were able to eliminate two large bin-ders of reference material, saving duplication costs and saving the students from lugging around course reference materials.

Electronic versions of reference material also enables students to have access to the most current version of the material, rather than using hard copies of material, which can become dated.

Another benefit was our ability to monitor internal and external Web page usage. We can easily view what organizations are currently logged on to the page. This gives us some insight into the magnitude of SPI internally and externally to our organization. We also use a shareware program to keep statistics on the Web page activity located at http://sepo.spawar.navy.mil/webstat.html. Over a given period of time we can see what organizations have logged onto the page and what products have been downloaded. This information helps us determine the priority of our product updates based on usage. Finally, by monitoring our Web page usage, we have verified that software process improvement is truly a worldwide phenomenon touching a wide variety of organizations from government, to banking, to insurance companies, to utilities.

Web Page Comments

We have been online for more than three years and have received more than 50,000 hits, which is respectable for such a specialized site. We have received hits from all over the world, from Argentina to Thailand, and many favorable comments. Here are a few examples from members of the software industry external to our organization:

> *"I have been using your home page for a while now and still find the information available, including links, fascinating reading. Your home page is the most comprehensive site I have found that is CMM®. I promote the fact that a Navy organization maintains the best site. Thank you for a job well done and please continue to support the home page."*

> *"This site is the most valuable site I could find on software development. The amount of documentation made available on software process is unbelievable. I could practically find all the documentation I needed to set up an organized development team, handle software projects, and improve such projects. Please continue to keep this site in existence since it is of utmost value to the developers who wish to enhance their knowledge of the software process."*

> *"The SEPO Web page is a truly superb product! Excellent information, excellent links to related and backup information. Indeed, it is one of the best and most content-laden software engineering Web pages anywhere. No one practicing software engineering in a DoD (or even U.S. government) milieu should be unaware of it."*

> *"I have been tasked to develop a 'standard' cost estimation process for (my organization). I intend to base this on the Software, Size, Cost, Schedule, Estimation Process document I downloaded from your Web site. I will include in the forward to our document credit to your command and team that developed the original document. I intend to 'adapt freely' your product to (my organization's) problem domain as an example of process improvement . . . I will provide a draft copy for your review when it becomes available. Thank you!"*

> *From the March 1997 issue of CrossTalk: "[This home page] has links to many major DoD and industry software policy and engineering home pages. It makes a great starting point to explore what is available."*

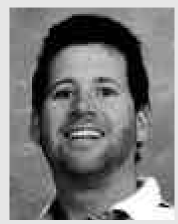
Summary

The SEPO home page represents the culmination of our SPI efforts to date. We encourage organizations embarking on SPI efforts to review our page and take freely any of the products that interest them. However, we ask that you provide us feedback on our products so we are able to continuously improve upon them—that is what SPI is all about. We encourage other SPI organizations to share their products and information with the software engineering community so that we all may improve on our efforts. Several organizations are already doing this, and we have used and appreciate their products, but more organizations need to contribute to this effort. Working together, we can make SPI a reality in every software engineering organization. Whether you are just beginning your journey, looking to fill in a few missing pieces in your program, or are interested in improving your processes based on what another organization is doing, we hope that our Web-based SPI repository will help you meet your needs.

Note

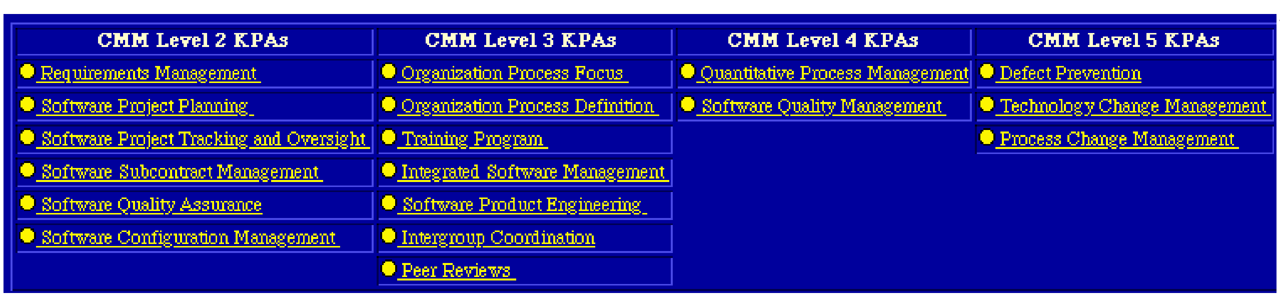
1. We hope readers will use some of our products and provide feedback.

About the Author



Brian Groarke has been a member of the Space and Naval Warfare (SPAWAR) Systems Center, San Diego Software Engineering Process Office (SEPO) for the past five years and has more than 16 years experience with the Navy. Before coming to SEPO, he worked on several software projects as a team member and software project manager. He is one of the key instructors for the SEPO Software Project Management course and is the developer and maintainer of the SEPO Web page. Groarke holds a bachelor's degree in computer science from Purdue University.

Software Engineering Process Office D12
SPAWAR Systems Center
53560 Hull St.
San Diego, Calif. 92152-5001
Voice: 619-553-6248 DSN 553-6248
Fax: 619-553-6249 DSN 553-6249
E-mail: groarke@spawar.navy.mil



CMM Level 2 KPAs	CMM Level 3 KPAs	CMM Level 4 KPAs	CMM Level 5 KPAs
Requirements Management	Organization Process Focus	Quantitative Process Management	Defect Prevention
Software Project Planning	Organization Process Definition	Software Quality Management	Technology Change Management
Software Project Tracking and Oversight	Training Program		Process Change Management
Software Subcontract Management	Integrated Software Management		
Software Quality Assurance	Software Product Engineering		
Software Configuration Management	Intergroup Coordination		
	Peer Reviews		

Visit this site on the World Wide Web at http://sepo.spawar.navy.mil

Figure 2. *Most SPI products are organized by KPA.*

SCM: More Than Support and Control

Software Configuration Management (SCM) supports people, controls data, and provides data integrity to baselines. Advances in automated tools and how baselines are created, maintained, and delivered requires careful, detailed planning. To be effective, SCM must be involved in day-to-day evolving, developmental, and ongoing maintenance activities.

Depending on where you go and whom you ask, you will always get different answers to the question, "Is SCM a *support* or *control* organization?" Even if you ask two people working on the same program, you likely get different answers. Those who are afraid it is a trick question, or who want to appear profound, will respond, "Why, it is both, of course." But when asked what they mean, they usually have other matters to attend to.

The answer is that SCM is a support *and* control organization, and when it is handled properly, a third factor is drawn into the picture, that of being a *service*.

If you talk with software developers, they likely will tell you they want support and some measure of control, but not too much. Management will generally say controls are a more important issue, as long as they do not impact costs and schedules. A SCM person with 10 or more years of experience, probably a first-line supervisor or manager, will agree with management, but will insist that sometimes cost and schedule must be impacted. A SCM person with fewer years under his or her belt will generally side with the software developer and strive to do whatever is necessary to assist him and get the job done.

The question becomes, "How can SCM best accomplish both support and control issues and be of service while adding value to the program effort?"

Support: SCM is a support organization in that it supports program engineers and developers, the program, the corporation, and in many situations, the customer.

Control: SCM is a control organization in that it controls specifications, documents, drawings, requirements, tools, software, and other deliverables.

Service: SCM is a service provider in that it *supports people* and *controls data*. This simple sentence is the primary key to a successful SCM operation.

The staff must be able to wear two different hats: one to support people, and one to control data. When those two hats get mixed up, i.e., SCM trying to control people and what/how they do things, problems and bottlenecks appear on the horizon. When this happens, often SCM is bypassed for the sake of "get the job out the door and we will fix it later."

It becomes the SCM manager's responsibility to:

- Ensure SCM personnel are properly trained and have necessary resources (budget and tools) to do an efficient and effective job.
- Ensure that a proper balance of control and support is tailored to each program that is supported.
- Ensure the SCM function is flexible and can accommodate changing needs and requirements of developers, customers, the program, and the company.

Future Directions

The SCM task has not changed much during the last 20 to 30 years. However, the environment within which SCM operates has significantly changed; that is likely to continue. Software language bases have changed: from Basic, COBOL and FORTRAN, to Ada and Pascal, to C++, Java, and many others. But that has not been the real impact to SCM; after all, code by any other name is still code.

More significant impacts to SCM have centered on automated tools and the library systems upon which they operate.

The tools have progressed from version control and semiautomatic build operations to systems that can establish and monitor the entire software development and production environment. Tools are more sophisticated and suppliers more numerous. Not long ago it was a somewhat simple matter for SCM to determine the best tool for the job. But in today's market, new issues have to be addressed before a decision is made. It is becoming increasingly important for representatives from each department within engineering organizations to consider, evaluate, and weigh their requirements against capabilities of various available tools.

SCM automated tools available today, and those on drawing boards, are much more versatile than their forerunners. But when asked, "Is there not one tool that will do it all?," the answer is still *no*. It is largely due to the fact that the SCM operating environment is still evolving.

In the recent past, SCM dealt with code and a few documents tucked away into a baseline where they could be easily controlled. With the introduction of Web-based repositories and increased involvement with commercial off-the-shelf vendors and subcontractor applications, baselines, as originally defined, are quickly becoming things of the past by becoming conceptual in nature. After all, when was the last time you *saw* a functional, allocated, or product baseline? Probably when you were dealing with hard copy documents and printed program lists. In the electronic office, these are nowhere to be found. The tendency is becoming to refer to the current version of controlled entities (code, documents, requirements, etc.) as the baselined version; previously, controlled versions were starting to be referred to as archived baseline versions.

In the past, most programs operated with three baselines: functional, allocated, and product (or requirements, design, and product, or some other set of descriptive labels). NASA once tried a system with nine baselines; it had a short life span. NASA, and a majority of other developers, was operating on the assumption that it was important to know into which baselines controlled items were placed. The primary concern was that the item was baselined, in that it could not be changed without a formal process, and without giving a name to the electronic partition where the item resides.

What drives this move away from the three former baselines? It is SCM and other developers raising questions such as, "Where do I put this .JPG file that is used in different documents?," "Should build scripts and make files also be controlled?," and "Where do we control corporate assets known as or to be captured as reusability or re-engineering issues?"

The questions are becoming, "Does the control board control the document or the information it conveys?" and "Does the control board control the software code or what it does? How does it do it, and and what is used to create it?" In the past, SCM controlled code and at times, documentation. What can be baselined in the environments with which we are now beginning to deal? The easier question is "What cannot be baselined?" Answer: SCM can baseline anything the program needs to control/make available. Answer: SCM controls data in any form so it can support people and provide an integral service to the program.

Lessons Learned

With only a few exceptions, if you look at any of the SCM standards, manuals, guides, books, etc., you will likely find that SCM has four major functions:

1. Identification
2. Change Control
3. Status Accounting
4. Audits and Verifications.

In nearly every case, *planning* is left out. And yet, SCM is using much more complex equipment to establish and maintain complex environments, multiple baselines, multiple environments on multiple platforms, etc. Like everyone else, SCM has to do this faster, cheaper, smarter, and better than before. Planning has become more important than ever. As recently as 10 years ago, there was still some truth to the statement, "Have them do SCM. Someone has to do it and anyone can learn it quickly enough." That is no longer true. The job has become too technical, too complex, and dependent on too many different variables to make it an easy job that anyone can pick up.

It is true that SCM still relies fairly heavily upon on-the-job training. No universities or colleges offer a four-year program in SCM. However, academia has recognized the evolving complexity of the job. Some two-year community colleges offer SCM certification programs. In 1998 I worked with three people who were pursuing doctorate degrees and centering their respective theses on SCM and its functions.

If this is all true, then *planning* cannot be interpreted as "A SCM plan has been written." That is a good start, but much more than a document explaining SCM's roles and responsibilities is needed. SCM planning activities must also include:

Metrics—how long, how many, when and where?

Skill Mix—what is needed and who has it or who can get it?

Infrastructure—who is doing what, where, when, how?

Contingencies—if this happens, then what?

Effort Tracking—manpower levels, roll on, and roll off.

Subcontracts—responsibility and authority.

Resources—budget, tool licenses, training, head count.

Matrix Management—decentralized workforce.

Control Transitions—informal to formal to field.

Records Retention—what gets kept, where, and for how long?

Control—who controls what and how do they do it?

Process—standardized procedures for repeatability.

Case Study 1

Last year, at a large aerospace corporation in the southeast, the SCM manager recommended the purchase of an automated SCM system that would satisfy all requirements laid out by SCM groups. Management placed the recommendation on hold to give other engineering departments time to review the tool. In the end, the recommendation to purchase it was cancelled. While the tool supported the SCM organization, it did not adequately address other developmental considerations the engineering ranks thought were important. Sometime later, a different tool was purchased that satisfied all the major requirements of SCM, software developers, software quality assurance, test, integration, and management organizations.

Case Study 2

During a recent visit to a private sector corporation (one that did not deal with government contracts) in New England, it was discovered that the developers' major concern about implementing SCM activities was so-called restrictions that would have to be addressed. They had been led to believe that SCM meant formal controls, restricted access, limited ability to apply creative solutions, and so on. When it was suggested that data can transition to formally controlled baselines through a series of informal control steps, and that SCM did not mean a lockdown or bottleneck, they became eager to be involved. After a number of meetings, a phased approach to formal SCM allowed for the placement of informal controls and data gathering, which led to baselined items. Everyone was pleased with the process. The developers soon realized they could work as a team with SCM to solve problems rather than as two separate organizations with their own concerns and desired solutions. More importantly, the SCM group learned that when it got out of its corner office and on the engineering floor (being support- and service-oriented) it quickly became an integral part of the engineering and development process and team.

Conclusion

In both of the above cases, SCM and developers soon realized they could work as a team to solve problems rather than as separate organizations with their own concerns and desired solutions. World class SCM operations can only be realized after properly planning for the implementation of SCM's four major functions. While SCM automated tools and baseline applications have become more and more complex, the fundamentals of SCM have taken on a new slant. SCM supports people, controls data, and provides a baseline integrity service. If that is not true where you work, it is time to ask why not.

About the Author

Bruce Angstadt is a Software Process Improvement Manager at Xerox Corp. in Webster, N.Y. With 35 years of government and industry experience in engineering support disciplines, his specific areas of interest are technical training and software configuration management. Prior to Xerox, he was employed by the Navy, Bell Laboratories, and Harris Corp. He also teaches a number of courses in configuration management for Systems Technology Institute of Malibu, Calif. He has a bachelor's degree in psychology and business management from Rollins College in Orlando, Fla.

Xerox Corp.
800 Phillips Road
Webster, N.Y. 14580
Voice: (716) 265-8700
Fax: (716) 422-0780
Email: Bruce.Angstadt@usa.xerox.com
Internet: http://www.scmtoday.com

Human Nature Has Not Changed

The collection of fables attributed to Aesop, but having other numerous origins, find application in many of today's working environments. The old adage "some things never change" holds true.

Fables have long been used to reflect the truths around us. Kenneth Grahame suggests fables have their "roots in the fixed and firm refusal of the community from its very beginning, to allow any of its members to go about calling any other one a fool or a rough, 'of his own mere notion'" [1]. Considering the popularity of Dilbert, for better or worse this attribute is not characteristic of present culture. However, these same fables provide us with vivid metaphors of the things we experience.

Little about Aesop's history is known. He was a slave, was eventually given his freedom, and may have spent time in the court of King Croesus (500-600 B.C.). It seems probable Aesop was not the author of all the fables commonly attributed to him. He most likely told versions of stories already in existence and may have added his own. History indicates the name Aesop became synonymous with the allegorical stories found in many cultures and societies.

Following are several Aesopian fables. In each, there are applications to software engineering, process improvement, project management and other environments. The more frequently we see and hear the messages, the more we are able to remember and apply the messages.

The Fox and the Hare

One day a hound, out hunting by himself, flushed a hare from a thicket and gave chase. The frightened hare gave the dog a long run and escaped. As the disappointed hound turned back toward home, a passing goat herd jeered, "You are a fine hunter! Are you not you ashamed to let a little hare one-tenth your size give you the best of it?"

"You forget," replied the hound, "that I was only running for my supper, but the hare was running for his life!"

Application: Organizational and personal survival is the greatest motivator. We must avoid developing a crisis environment. However, clearly communicating the relationship between individual actions and survival is a powerful tool.

The Mice in Council

For many years the mice had lived in constant dread of their enemy, the cat. They decided to call a meeting to determine the best way to handle the situation. Many plans were discussed and rejected.

At last a young mouse got up.

"I propose," said he, looking very important, "that a bell be hung around the cat's neck. Whenever the cat approaches, we always shall have notice, and so be able to escape."

The young mouse sat down amidst tremendous applause. The suggestion was put to a motion and passed almost unanimously.

Just then an old mouse, who had sat silently all the while, rose to his feet. "My friends, it takes a young mouse to think of a plan so ingenious and yet so simple. With a bell about the cat's neck to warn us we shall all be safe. I have but one question to put to the supporters of the plan—which one of you is going to bell the cat?"

Application: Planning is one thing, execution is another. Plans must be based on realistic expectations.

The Birds, the Beasts, and the Bat

Once upon a time, war broke out between the birds and the beasts of the earth. For a long while the issue of the battle was uncertain. The bat, taking advantage of the fact that he had certain characteristics of both, kept aloof and remained neutral.

The birds said, "Come with us." But he shook his head and said, "I am a beast." Later some of the beasts of the earth approached him and asked him to join their side. He refused. "I am a bird," said he.

In due course, peace was concluded between the embattled birds and beasts. The bat flew blithely up to the birds to join them in their rejoicing. But the birds gave him the cold shoulder and flew away. The beasts did the same. Condemned by both sides and acknowledged by neither, the unhappy bat skulked away to live in holes and corners, never caring to show his face except in the dusk of twilight.

Application: Playing both sides against the middle often leaves you with nothing. A lack of commitment is a primary cause of failure. Failing to commit is committing to fail.

The Three Tradesmen

The enemy stood outside the walls of a certain city. As the soldiers brought up their siege weapons and arranged their forces for the attack, the desperate defenders within held a council of war to determine the best means of holding the city.

A bricklayer arose,

"Sirs," said he, "it is my opinion that the best material for the purpose is brick." Then he sat down.

A carpenter asked to be recognized.

"I beg to differ with the bricklayer. The material that best serves our desperate needs is wood. Let timber be our defense!"

Then the tanner jumped to his feet.

"Citizens," he cried, "when you all have had your say, I wish to remind you that there is nothing in the world like leather!"

Application: Sometimes we are so in love with our way of doing things, we can not see beyond our noses.

The Fisherman Piping

There once was a fisherman who enjoyed playing the bagpipes as much as he did fishing. He sat down on the riverbank and played a merry tune, hoping that the fish would be attracted and jump ashore.

When nothing happened, he took a casting net, threw it into the water, and soon drew it forth filled with fish. As the fish danced and flopped about in the net, the fisherman shook his head.

"Since you would not dance when I piped, I will have none of your dancing now."

Application: Doing the right thing at the right time is a great art. Someone has said knowledge is knowing what to say. Wisdom is knowing when to say it.

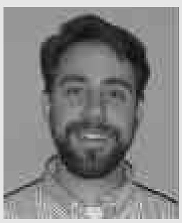
So What?

These parables show that in the history of mankind, human nature has not changed much. The issues, problems, and challenges people have historically confronted are still our nemeses today. However, if we are mindful of the pitfalls that are capable of ensnaring us, we may better avoid them.

References

1. *Fables from Aesop*, Ennis Rees, Oxford University Press, New York, N.Y., 1966.
2. *Aesop's Fables*, Grosset and Dunlap, New York, N.Y., 1963.

About the Author



Jeremiah Smedra is a Proposal Manager with General Electric Energy Services. He manages the bid process for field service supplied to power providers in the western United States. As a former consultant with the Software Technology Support Center, he provided support to Department of Defense organizations pursuing process improvement and project management education. Smedra has a bachelor's degree in Marine engineering systems from the U.S. Merchant Marine Academy in Kings Point, N.Y. He is a registered EIT and a Project Management Institute-certified Project Management Professional.

General Electric Energy Services
2180 South 1300 East, Suite 340
Salt Lake City, Utah 84106
Voice: 801-468-5712
Fax: 801-468-5767
E-mail: jeremiah.smedra@ps.ge.com

Coming Events

March 6-8
13th Conference on Software Engineering Education and Training
http://www.se.cs.ttu.edu/CSEET2000

March 6-10
Software Management/Applications of Software Measurement
http://www.sqe.com/smasm/2000/

March 13-17
6th International Conference on Practical Software Techniques
http://www.softdim.com

March 20-22
5th Annual Association for Configuration & Data Management Technical Conference
http://www.acdm.org

March 20-23
12th Software Engineering Process Group Conference
http://www.sei.cmu.edu/products/events/sepg/

April 11-14
Infosecurity 2000
http://www.infosec.co.uk/page.cfm

April 15-18
ACM International Conference on Management of Data
http://www.seas.smu.edu/sigmod2000

April 18-20
FOSE: Leading-Edge Technology for Leaders in Government
http://www.fedimaging.com/conferences

April 24-28
SEA 2000
E-mail for information: johnl@sea-act.com.au

See p. 30 for information on STC 2000, April 30-May 5

Quote Marks

"The real question is not whether machines think but whether men do. The mystery which surrounds a thinking machine already surrounds a thinking man."
– B.F. Skinner, Contingencies of Reinforcement

"Man is a game-playing animal and a computer is another way to play games."
—Dilbert cartoonist Scott Adams

"A computer terminal is not some clunky old television with a typewriter in front of it. It is an interface where the mind and body can connect with the universe and move bits of it about."
—author Douglas Adams, Mostly Harmless

"Technology makes it possible to gain control over everything, except over technology."
— John Tudor

"Software. These programs give instruction to the CPU, which processes billions of tiny facts called bytes, and within a fraction of a second send you an error message that requires you to call the customer-support hot line and be placed on hold for approximately the life span of a caribou."
–Dave Barry, "Get With the Program," Reader's Digest

"Of all software products, the highest percentage of reuse is in the proposal."
—Samuel T. Redwine Jr.

Join Us for An Incredible First Week of May at STC 2000

STC 2000 is the premiere Department of Defense Software Technology Conference, co-sponsored by the Department of the Air Force, Department of the Army, Department of the Navy, and the Defense Information Systems Agency (DISA). Utah State University Extension is the nonfederal conference co-sponsor. More than 3,500 participants from the services, other government agencies, contractors, industry, and academia are expected to attend April 30-May 5 in Salt Lake City, Utah.

"Software and Systems—Managing Risk, Complexity, Compatibility and Change" is the theme. Information used during the next millennium will require systems and software interoperability. This interoperability must be across all services and fighting forces.

General sessions include speakers you will not want to miss:

Opening General Session–May 1

Government Keynote Address by Dr. Delores Etter, Deputy Under Secretary of Defense (Science & Technology);
Industry Keynote Address by Paul Maritz, Group Vice President, Developer Group, Microsoft Corp.;
Academia Keynote Address by Dr. Roger Firestien, Associate Professor, New York College at Buffalo.

Closing General Session–May 4

Keynote address by Jon "maddog" Hall, Executive Director, Linux International.

Individual service sessions immediately follow the opening general session. These sessions, held by the conference co-sponsors, are open to all attendees, and are similar to a 'commanders call.' Lt. Gen. William Campbell will host the Department of the Army meeting, Dr. Donald Daniel will host the Department of the Air Force meeting, and Rear Adm. Kenneth Slaght will host the Department of the Navy meeting. Each session includes a question-and-answer session.

Conference co-sponsors have again agreed to participate in a question-and-answer general session on May 2. They will address questions regarding appropriate issues submitted prior to and during the session.

Attendees have a difficult choice for lunch on May 1. There are three concurrent speaker luncheons for which attendees can preregister. Luncheon Keynote Addresses will be given by:

Maj. Gen. John Campbell, Vice Director, Defense, DISA, Commander, Joint Task Force, Computer Networked Defense;
David Richwine, Maj. Gen., USMC (Ret.), Executive Vice President, Armed Forces Communications and Electronics Association;
Jay H. Nussbaum, Executive Vice President, Oracle Service Industries.

Attendees should indicate on the registration form which luncheon they want to attend. The luncheon is included in the conference fee.

New this year, a book signing will be held May 3 in the Exhibit Hall. Many speakers have written books on subjects relating to topics of their presentations. A number of them will sign their books. Attendees will be able to purchase books at an exhibit-area bookstore.

The Institute of Electrical and Electronics Engineers is presenting a special one-day workshop May 5 for all attendees at no extra charge. The theme is, "Guide to the Software Engineering Body of Knowledge (SWEBOK): Overview and Applications," a project developing a broad consensus on the core contents of the software engineering discipline. The guide may be used in developing software engineering curricula, accrediting curricula, developing licensing examinations, and describing and certifying competencies.

"Plan to be in Salt Lake City to explore new software ideas and trends at this premiere conference. Software experts from government, industry, and academia will explore new ideas and technologies for information systems that will be used during the new millennium."

—Lt. Gen. David J. Kelley, Director, DISA
—Lt. Gen. William Campbell, Director of Information Systems for Command, Control, Communications and Computers—U.S. Army
—Dr. Donald C. Daniel, Deputy Assistant Secretary of Science, Technology, and Engineering—U.S. Air Force
—Rear Adm. Kenneth D. Slaght, Chief Engineer, Space and Naval Warfare Center—U.S. Navy

It's not too late to register!

You can still register online through our secure Web server at http://www.stc-online.org/, via fax with credit card billing information or a copy of a completed purchase order, 435-797-0636 or 435-797-0036, via telephone with credit card billing information, 800-538-2663 or 435-797-0423, or register in person. Credit cards will not be charged until April 10.

Visit our Web site at www.stc-online.org for conference and exhibit information, registration forms, and housing information. If you would like a copy of the registration brochure sent to you, please send an e-mail request to lynne.wade@hill.af.mil or call Lorna Baker at 435-797-0039.

Please contact us if we can be of further assistance. This is one conference that you do not want to miss. See you in May!

Dana Dovenbarger, Conference Manager
Lynne Wade, Assistant Conference Manager
Software Technology Support Center
OO-ALC/TISE
7278 4th St.
Hill AFB, Utah 84056-5205
Voice: 801-777-7411 DSN 777-7411
Voice: 801-777-9828 DSN 777-9828
Fax: 801-775-4932 DSN 775-4932
E-mail: dana.dovenbarger@hill.af.mil
lynne.wade@hill.af.mil

CMMI "Just-in-Time" Training CD-ROM offered to STC Conference attendees

Working in collaboration with the Software Engineering Institute to help organizations transition to the use of the Capability Maturity Model® (CMM)® Integration (CMMI SM), the Computer Resources Support Improvement Program (CRSIP) is sponsoring and funding an effort to demonstrate the utility and feasibility of using Just-in-Time CD-ROM-based training with access to the Internet. A CMMI CD-ROM is being developed, for distribution to all attendees of STC 2000, that demonstrates the use of the technology in providing an alternative CMMI training and information delivery scheme. Since STC 2000 precedes the release of CMMI v. 1.0, draft material will be used. Anyone who offers review and feedback comments on the presentation alternatives and use of the technology in supporting the CMMI transition and training will be offered a free CD-ROM update after release of CMMI v 1.0.

Software Engineering Certification

The IEEE Computer Society and the ACM approved a Software Engineering Code of Ethics and Professional Practice to provide a standard for teaching and practicing software engineering. http://computer.org/en/tab/swecc/code.htm. Many states are considering laws similar to that in Texas that requires certification of individuals claiming to be a "software engineer." Because of the topic relevance, IEEE will sponsor a special track during Software Technology Conference in May 2000 to discuss software engineering best practices, standards, education and accreditation.

Give us your information–get a free subscription

Fill out and mail this form to us for a free subscription to CROSSTALK.

OO-ALC/TISE 7278 FOURTH STREET, HILL AFB, UTAH 84056

Or save a stamp and subscribe online at http://www.stsc.hill.af.mil/request.asp

FULL NAME:______________________________

RANK OR GRADE:______________________________

POSITION OR TITLE:______________________________

ORGANIZATION OR COMPANY:______________________________

ADDRESS:______________________________

____

BASE OR CITY:__________ STATE:________ ZIP:________

VOICE: COMMERCIAL__________ DSN:__________

FAX: COMMERCIAL__________ DSN:__________

E-MAIL: __________@__________

1999 BACK ISSUES AVAILABLE (CHECK THE BOX OF THE MONTH(S) DESIRED.)

JANUARY ☐ FEBRUARY ☐ MARCH ☐ APRIL ☐ MAY ☐ JUNE ☐

JULY ☐ AUGUST ☐ SEPTEMBER ☐ OCTOBER ☐ NOVEMBER ☐ DECEMBER ☐

For more information about how the STSC can help your organization buy and build software better, please visit our Web site at http://www.stsc.hill.af.mil

Beware the Ides of March Madness

BACKTALK

This time of year, it can be difficult to concentrate on anything but basketball, specifically the NCAA Men's College Basketball Tournament. If you have accrued a couple of extra sick days, you can watch the opening rounds.

The CBS SportsLine® streaming real-time scoreboard with moving X's and O's and detailed statistical information can help you "work" more efficiently during March, providing you work where such tantalizing items make it past the firewall.

However, when an avid basketball fan is deprived of the opportunity to watch every minute of every game (sometimes in a split screen) of the tournament, it can result in a temporary psychotic break that causes this fan to free-associate numbers of the tournament with numbers of software and ~~hardwood~~ hardware.

This tournament is a single-elimination event that is the epitome of fairness: anyone can win. All games are played on neutral courts, and anything can happen. School Never-Been-There-Before is capable of beating School Been-There-Every-Year. In a way, it's like the software market.

Each year, 64 teams make it to the so-called Big Dance. It used to be 32 in the good old days and some say it should now be 128 so every (basketball) program is given a chance to run. These numbers remind me of RAM.

Now, 64 MB of RAM is not enough to efficiently run a program like Quark Xpress, but since there are only 64 teams in the tournament, it suffices.

After the first tournament round, 32 teams remain. That makes me think of all the 32-bit programs I can install. Those 32 teams play against each other and are whittled to a so-called Sweet 16. And those 16-bit programs were sweet in their day.

The (basketball) programs take a few days off before moving to the regional finals, where 16 teams are pared down to eight. Some call them the Elite Eight; others refer to them as to the Super Eight. Whatever you call them, you definitely need at least 8 GB of storage space to hold all that software.

The winners of the eight become the Final Four, the finest teams in the land. You do not have to be a basketball fan to be acquainted with the *hoopla*. By the way, we are in the fourth generation of computer systems with an eye on the fifth with implications of natural languages and artificial intelligence. The fifth generation will be akin to a fifth wheel. How could we want anything more than four?

Duke and Utah will defeat Cincinnati and North Carolina to become the final two teams (OK, so maybe I won't win my office pool . . .) and the final two contestants in the computer realm will be the human being and the computer.

Am I a cynic if I can imagine that the computer will be the last one standing?

Wait a minute. While this enormous Quark file was freezing up my computer, Randy the systems guy told me he would install 64 more MB of RAM. I'll need to go back and add some more ~~buzzers~~ bells and whistles to this issue. And please understand I wrote this from a ~~layupman's~~ layman's perspective.

—Matt Welker

Dr. Delores Etter, Deputy Under Secretary of Defense (Science & Technology) will deliver the Government Keynote Address. Dr. Etter is a frequent contributor to *CrossTalk* (see p. 5). Complete biographical information is available at www.dtic.mil/ddre/bios/etter.html

STC 2000

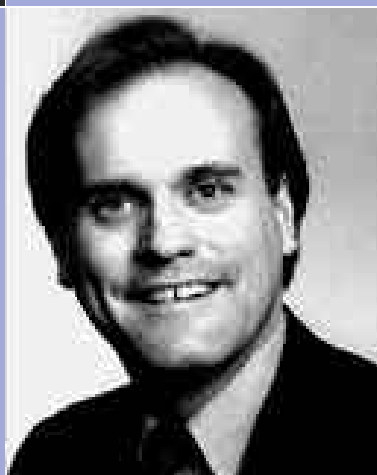
"Software and Systems— Managing Risk, Complexity, Compatibility and Change"

April 30-May 5

Salt Lake City, Utah

Opening General Session Speakers

Roger Firestien, President of Innovation Systems Group, associate professor at the State University of New York College at Buffalo, adjunct professor of leadership and public policy at Gannon University in Erie, Pa., will deliver the Academia Keynote Address.



"Roger Firestien is the gold standard of creativity training. He has changed the way our whole work force looks at problems. Creativity doesn't have to be a mystery. [He] gives you practical methods to unleash your creative power."

—The Clorox Company

There is still time to register!

See p. 30 for information.

Paul Maritz, Vice President of Microsoft® Developer Group, will deliver the Industry Keynote Address. Maritz is also a member of the Microsoft Business Leadership Team, a group that shares responsibility (with President and CEO Steve Ballmer) for strategic and business planning.

Published by the Software Technology Support Center

S T S C

Sponsored by the Computer Resources Support Improvement Program (CRSIP)

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

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