The DoD Software Tech News

FAA

STN Vol. 4, No. 3

FAA Information Technology Initiatives
pg 2

For the FAA, Software Process Improvement is Not Enough
pg 5

Toward Defining a Software Engineering Body of Knowledge
pg 8

Application of Software Assurance Best Practices to CNS & ATM
pg 15

Building Trustworthy Systems: An FAA Perspective
pg 18

Unclassified and Unlimited Distribution

An Introduction to FAA Information Technology Initiatives

Introduction

The Federal Aviation Administration (FAA) operates the Nation's air traffic control system and regulates aviation safety, security and the U.S. commercial space industry. In its position on the front line of aviation safety, the FAA works with the air transportation industry, other agencies at the Federal, state, and local level, the academic community, and with its international counterparts. The goal of this wide-ranging collaboration is to provide a technically advanced airspace system that meets the highest attainable levels of safety, security and efficiency.

Information technology plays a central role in all areas of the FAA mission from the management of the safe and efficient flow of aircraft through the nation's airspace, to carrying out aggressive security measures to protect air travelers from acts of terrorism and criminal misconduct to the speeding the transition of the next generation of air traffic control technology. Information technology provides the core technology from which all aspects of the FAA's mission are accomplished. It is without question, that the FAA has been transformed into an information-centric agency where improvements in the IT infrastructure are necessary to meet the ever increasing requirements and demands of increased growth in air travel and transport throughout the world.

Lines of Business

The FAA team now includes almost 50,000 employees concentrated in six strategically focused lines of business (LOBs). These include Air Traffic Services (ATS), Regulation and Certification (AVR), Civil Aviation Security (ACS), Airports (ARP), Research and Acquisitions (ARA), and Commercial Space Transportation (AST).

ATS provides an integrated set of services to ensure that each aircraft operation is safe from the time pilots begin pre-flight activities until they shut down aircraft at their destination. Air traffic controllers at local airports direct airplanes that are taking off, landing, or flying within the visual range of their tower. Controllers in terminal radar control (TRACON) facilities handle aircraft for one or more airports in a large metropolitan area. Controllers at 21 air route traffic control centers (ARTECC) guide airplanes in flight from one city to another. Flight Service Stations (FSS) provide flight plan filing, weather data, and aeronautical information briefings to pilots. On a typical day, FAA controllers will handle over 200,000 takeoffs and landings, moving almost 2 million passengers. Services are available 24 hours a day, 365 days a year. In 1998, the FAA established the Free Flight Phase 1 (FFP1) program to speed the deployment of automation and decision making tools that will increase systems safety and capacity and bring benefits to users in the form of fuel and crew cost savings. These include tools to aid controllers in aircraft sequencing, conflict detection, and collaborative decision making. These tools are being introduced incrementally at selected locations. As can be seen, the FAA's Air Traffic Service LOB is one of the largest, most complex and critical infrastructures developed and managed by the Federal Government.

Regulations and Certification (AVR) oversees the safety of planes and the credentials and competency of pilots and mechanics, develops mandatory safety rules, and sets high standards for civil aviation. Each year the FAA performs more than 347,000 inspections and investigations and takes approximately 12,000 enforcement actions, helping to make air travel among the safest modes of transportation available today. Two on-going efforts, Safer Skies and the Air Transportation Oversight Systems (ATOS) are designed to raise the bar on safety. Safer Skies is a focused data-driven program that attempts to find the root cause of accidents and then targets resources to find and apply the right interventions. ATOS complements Safer Skies and provides a new oversight approach that uses system safety principles and systematic processes to assure that air carriers have safety built into their operating systems. ATOS will allow inspectors to quantify and document both air carrier system weaknesses and best practices. This will allow inspections to be targeted to specific air carrier requirements and will increase the value of information provided from surveillance. ATOS will lead to a more collaborative partnership with the air carriers to identify and correct safety related issues.

The Civil Aviation Security LOB works with local, national and international security, intelligence, and law enforcement agencies to protect passengers, personnel, aircraft, and critical national airspace facilities against terrorist and other criminal acts. Threats are monitored continuously and, when necessary, the FAA orders heightened precautions. Current R&D in the development of new and innovated trace detection and explosive detection devices are on-going with vendors from across the world.

The Airports LOB, like the airlines, are vital links in the air transportation network. The FAA works in partnership with airport authorities, local units of government, metropolitan planning organizations, and states to revitalize and expand the Nation's airports. As part of its safety oversight mission, the FAA certifies airports serving air carrier operations and inspects those airports for compliance with established safety criteria.

by Marshall R. Potter, Chief Scientist for Information Technology, FAA

The Research and Acquisitions LOB conducts research and invests in essential infrastructure to meet the demands for higher levels of safety, security, capacity and efficiency. Research priorities include explosive detection, weather, aircraft structures, noise mitigation, human factors, satellite navigation and information systems security. Additionally, the FAA is in the midst of several major acquisitions to modernize the Nation's air traffic control (ATC) system.

The Commercial Space Transportation LOB licenses commercial space launches and sites to protect public health and safety of property.

Articles

In the articles that follow, you will find initiatives of the Assistant Administrator for Information Services and Chief Information Officer of the FAA. Under the capable leadership of Dr. Daniel Mehan the Associate Administrator and Dr. Arthur Pyster his Deputy, the FAA has made rapid progress in improving its business practices and processes, improvement in its Information Systems Security (ISS) program and the definition of an Information Technology Architecture with major efforts focused on ISS, Data and Technical architectures. The Office of the Administrator for Information Services and Chief Information Officer consists of three Divisions (Information Analysis, Process Engineering, and Information Management), the FAA's Software Engineering Resource Center (SERC), the Office for Information Systems Security and the Office of the Chief Scientist for Information Technology.

The Information Analysis Division (AIO-100) is currently involved in implementing portfolio management as a technique to evaluate and select what will be in the FAA capital investment portfolio. This technique is a "best practices" technique currently in wide use among information rich industries to enable their management to make prudent investment decisions. Additionally, the Information Analysis Division is also implementing a new FAA Business Plan. Working with FAA's customers and the aviation industry, the agency's senior managers are crafting a business plan that is both business-oriented and reflects their vision of the future for aviation and space transportation. The plan's purpose is to align the agency's spending with its strategic imperatives.

The Process Engineering Division (AIO-200) is developing mechanisms to identify, enhance, and promote the adoption throughout the agency of the best applicable information technology processes, practices, and tools. In cooperation with lines of business, Process Engineering develops and helps implement plans that maximize the effective use of IT in meeting agency business needs. It focuses on activities that cut across lines of business with special emphasis on those activities that have the most impact on the agency's ability to meet its business goals. Three key projects of AIO-200 include;

- the FAA's Capability Maturity Model,
- work on Software Safety and Assurance,
- the development of a Software Engineering Body of Knowledge.

The first three papers that are included in this special issue of *DoD Software Tech News* provide more details on these three initiatives.

The Information Management Division (AIO-300) is focused in the following areas: Implementation of the Data Management Strategy, the development of agency Information Technology policy and plans, the Implementation of the FAA Metadata Repository, the implementation of an Internet monitoring program, the exploration of the opportunities for an agency-wide Directory Services supporting FAA's initiatives to establish a public key infrastructure (PKI) and the exploration of Information Technology opportunities that are agency wide and have the potential for significant return on investment (e.g. Next Generation Messaging Requirements).

The Office of Information Systems Security (AIS) provides policy, oversight and support to the FAA across the entire organization and has the difficult task to work with the LOBs to facilitate that adequate Information Systems Security (ISS) is integrated into new acquisitions and at the same time assure that ISS needs of current legacy systems are also provided. A fourth paper has been included in this special issue to address the FAA's approach to ISS and the meaning of trustworthiness as it is seen within the FAA.

The Software Engineering Resource Center (SERC), located at the William J. Hughes Technical Center in Atlantic City, NJ, provides the technical and engineering support to AIO for software engineering issues. Currently, the SERC is focused on developing an Adaptation Improvement Program for all aspects of the National Airspace System (NAS).

The Office of the Chief Scientist for Information Technology serves as a catalyst, information resource and leader of the FAA's IT R&D efforts. The Chief Scientist, working in close interaction with all the different components of AIO and the LOBs, is striving to enhance all IT aspects of the FAA.

The following articles provide a glimpse of the many on-going FAA IT initiatives. The article by Dr. Linda Ibrahim describes the FAA's approach

Continued on page 4

Continued from page 3

to integrated process improvement. The FAA developed the FAA-iCMM in 1997 to guide improvement of its software engineering, systems engineering, management, and acquisition processes in an integrated, effective, and efficient manner. Today, the FAA-iCMM is being used by all the LOBs and is the corner stone of all FAA process improvement efforts.

The article by Iraj Hirmanpour, Soheil Khajenoori, Thomas Hilburn, and Richard Turner discusses a major accomplishment in defining a Software Engineering Body of Knowledge. This effort was one of the first comprehensive efforts to define what software engineers need to know to do their jobs. The paper offers an overview of the comprehensive categorization that the SwE-BOK project has accomplished to date.

The article by Ronald Stroup on Application of Software Assurance best practices provides a glimpse into the FAA's software safety and assurance efforts as they apply to communications, navigation, surveillance and air-traffic management systems. The paper provides a summary of how software assurance and safety techniques are being integrated into the overall FAA system safety assessment process.

The final article, by Dr. Daniel Mehan, the FAA's CIO, and myself provides a glimpse at the role of Information Systems Security (ISS) and meaning of trustworthiness as it is viewed within the FAA. In this article, we define a holistic approach to ISS and provide multiple views into how the FAA is attempting to solve this critical problem. At the end of this article, we propose that the priority that you assign to your goals or functions, in this case the five levels of security defined by the cyber pyramid, not only impacts, but strongly defines the ultimate structure of the information systems security architecture (ISSA). Organizations such as the FAA that do not have strong needs for confidentiality and access control, as do many DoD organizations, may find more effective and efficient architectures for meeting their specific ISS needs that focus first on obtaining availability and integrity of the integrated systems.

I hope that you will enjoy reading about these FAA initiatives If you would like to know more about FAA IT initiatives and how AIO is attempting to improve the safety, security and efficiency of the airspace, I recommend that you contact the authors and look at the FAA AIO web pages at www.faa.gov/aio/

Biography

Marshall Potter is the Technical Advisor and Chief Scientist for Information Technology at the FAA working directly for the Associate Administrator for Information Technology and the Chief Information Officer (CIO). In this position, he is responsible for providing expert technical and scientific advice and guidance on all aspects of information technology with special emphasis on information security and assurance, information management, process engineering, and information architectures. In the area of Information Security and Assurance, he is currently directing the FAA's PDD-63 (Critical Infrastructure Protection) R,E&D program and is leading the FAA effort to use Common Criteria Protection Profiles in large system acquisitions.

Prior to coming to the FAA in March 2000, Marshall was the Special Assistant for Computing and Software Technologies in the office of the Director, Defense Research and Engineering (DDR&E) within the Office of the Secretary of Defense (OSD). In that position, Marshall provided strategic and day-to-day oversight on the DoD Information Technology Science and Technology Program covering a wide range of R&D areas. Marshall was also the author of the Information Superiority chapter of the Year 2000 Joint Warfighter Science and Technology Plan (JWSTP). Marshall was a recipient of the Defense Exceptional Civilian Service Award in 2000. Mr. Potter has over 30 years experience within the Federal Government.

Marshall is an Adjunct Associate Professor of Computer Science at the University College, University of Maryland in College Park, MD where he has taught courses in Operating Systems Theory, Computer Systems Architecture and Software Engineering for over 15 years. He did both his undergraduate and graduate work in Electrical Engineering and Computer Science at the University of Maryland. Marshall has been a member of the IEEE and ACM since 1970 and is also a member of Tau Beta Pi, Eta Kappa Nu, Omicron Delta Kappa and Phi Eta Sigma honoraries.

Contact Information

Chief Scientist for Information Technology
Federal Aviation Administration
800 Independence Ave, SW
Washington, DC 20591

Phone: (202) 267-9878
E-Mail: marshall.potter@faa.gov
URL: www.faa.gov

For the FAA, Software Process Improvement is Not Enough

by Dr. Linda Ibrahim, Federal Aviation Administration

Abstract

This paper describes the Federal Aviation Administration' s(FAA) approach to integrated process improvement and illustrates what an organization might do if it decides that software process improvement is not enough.

Introduction

Organizations like the FAA, seeking to improve the way they do business, frequently look to a variety of models and standards that provide guidance for improving parts of the enterprise. Capability Maturity Models® (CMMs®) have been a popular approach, especially in organizations whose activities include software engineering. But what about improving other core disciplines? Is software process improvement enough? If not, improvement initiatives directing separate implementation of multiple standards can cause confusion if the models overlap or are related, cross organizational or managerial lines, or are pursued in isolation without gaining the advantages of their use together.

This was the case in the FAA in 1996. At that time, three single-discipline CMMs were being used separately to guide process improvement: the CMM for Software [1], the Systems Engineering CMM [2], and the Software Acquisition CMM [3]. While some improvements were being made, the single-discipline CMMs were being used in an uncoordinated way and without much success. These CMMs have different architectures, goals, terminology, and appraisal methods; they entail considerable overlap; and none alone covers all FAA system life cycle activities. Thus improvement efforts were suboptimal and the goal of FAA-wide, full life cycle, cross-discipline process improvement remained elusive. In addition, the FAA had moved to using integrated product teams and these teams needed processes and process improvement guidance that interrelated their disciplines.

Thus in 1997, the FAA developed an integrated CMM, the FAA Integrated Capability Maturity Model (FAA-iCMM) [4] to guide improvement of its software engineering, systems engineering, management, and acquisition processes in an integrated, effective, and efficient way. The FAA-iCMM is the first major integrated CMM, and has provided proof of concept that CMM integration works.

The FAA-iCMM and Integrated Process Improvement

CMMs are repositories for best practices generally observed to be effective in government and industry. These best practices provide two types of guidance:

1. guidance for *performing* selected processes, and
2. guidance for *improving* process performance.

The single-discipline CMMs focus on best practice guidance for performing and improving processes in a single discipline such as software acquisition or systems engineering. The FAA-iCMM, by contrast, captures and integrates in a single model all principles and practices of the three single-discipline CMMs that FAA had been using separately: the CMMs for software, systems engineering, and software acquisition. The FAA-iCMM contains 23 process areas that integrate the 52 process areas and key process areas of its three source models. Each process area contains best practices integrated from the source models, and provides guidance for performing integrated processes. Best practice guidance for improving processes has also been integrated in the FAA-iCMM.

A single approach is applied across all processes, providing an integrated approach to process improvement. Capability levels are defined which provide a path for improving any process, and maturity levels group processes together to provide guidance on what to improve next. Thus, the FAA-iCMM helps an organization pursue integrated process improvement in two ways: by providing guidance for performing integrated processes, and guidance for improving those processes in an integrated way.

What is an Integrated Process?

There are several dimensions that may characterize integrated processes such as those found in the FAA-iCMM. An integrated process may be multidisciplinary, requiring participation of experts from more than one functional area or discipline. Examples include a requirements engineering process for a software-intensive system involving systems engineers and software engineers; an acquisition management process involving acquisition program managers and technical project managers; an outsourcing process involving procurement and legal experts, plus engineers. An integrated process may cross life cycle phases, for example, a transition process spanning development and maintenance, or a customer communication process performed during all life cycle phases. An integrated process may support multiple disciplines. Examples include: a configuration management process that manages work products

Continued on page 6

Continued from page 5

and baselines for systems, hardware, software, contracts, internal documentation, etc.; a human resource process that develops skills of practitioners in several disciplines; a quality assurance process that pertains to any processes and products. Lastly, an integrated process may cross an organization in several ways, for example, strategic planning, across an enterprise, or management applied to an enterprise, a division, a program, a project, or a task.

What is an Integrated Process Improvement?

In the FAA-iCMM, process capability levels and generic process improvement practices are integrated to provide a common approach to the improvement of any process. The same terminology is used, and organizational process improvement goals can be expressed in a common way across all disciplines by reference to the integrated model. Process capability is measured by consistently applied appraisals using the FAA-iCMM Appraisal Method [5], which integrates several appraisal methods and variations.

Deploying Integrated Process Improvement

Initial Deployment

The FAA-iCMM rapidly became the single framework for CMM based improvement in the FAA. In 1997, the Associate Administrator for Research and Acquisition (ARA) targeted selected major acquisition programs to achieve maturity level 2 on the FAA-iCMM by December 1999. Soon several additional programs and organizations across the FAA life cycle began applying the FAA-iCMM, including FAA's System Requirements Service organization. In 1999, ARA and the Associate Administrator for Air Traffic Services (ATS) committed to a common FAA-iCMM based process improvement goal to realize high quality solutions to Agency and user needs, predictable cost and schedule, and increasing productivity. FAA's operational support organization committed its engineering programs to FAA-iCMM based process improvement, and the Office of Independent Test and Evaluation adopted the FAA-iCMM.

Some Results

In 1999, the FAA Technical Center achieved its goal of capability level 2 in four FAA-iCMM process areas. Then in early 2000, an extensive full life cycle appraisal determined that several major FAA acquisition and engineering programs have achieved FAA-iCMM maturity level 2. In addition, capability level 2 in several process areas was uniformly achieved in programs appraised across the FAA life cycle. Early anecdotal data indicate that FAA-iCMM based improvement efforts have led to: better predictability, improved communication, improved teamwork, increased quality, greater consistency, cost savings, cost reductions, time savings, clarity of roles and responsibilities and processes, a more streamlined work effort, easier training of new hires, and more appreciation and attention to the roles of different disciplines and processes. First cost-benefit analyses have convinced the Agency that integrated process improvement provides value. More formalized quantitative measures are being established.

Next Steps

The FAA is continuing to deploy the FAA-iCMM across more and more programs and organizations in the agency. In addition, the FAA is evolving the FAA-iCMM in two ways:

1. upgrading current content to retain currency and include lessons learned from initial use, and
2. enhancing the model to include additional disciplines. The vision for the FAA-iCMM is that it is the model that provides guidance for enterprise- wide process improvement, integrating and relating and containing in one model current best practice guidance found in the most widely utilized and recognized models and standards adopted by government and industry, as applicable to the business of the FAA.

Is Software Process Improvement Enough for You?

Consider a few questions about software. What percentage of your organization's processes are software processes? What other processes affect software and your success in software projects? What if software processes are improved and others aren't? Do other processes need improvement? Consider whether integrated process improvement across your organization makes sense. Do you perform integrated processes? Do you have integrated teams? Are your improvement efforts supporting enterprise-wide needs efficiently and effectively?

If you decide to pursue integrated process improvement, it's not easy! Be prepared to face and resolve cross-organizational issues, establish an integrated process improvement infrastructure, and commit the energy and resources necessary to make it happen. At the FAA, the factors most critical in launching and sustaining

integrated process improvement have been the adoption of the FAA-iCMM to guide integrated process improvement, and performance goals tied to integrated process improvement to make sure it happens. (For more information on FAA's experiences in integrated process improvement, refer to [6] and [7].)

Lastly, the FAA-iCMM is not specific to the FAA and can be used by any organization that seeks to improve engineering, management, and acquisition in an integrated way. For example, enterprise-wide process improvement efforts have been piloted at Warner Robins Air Logistic Center using the FAA-iCMM, and the National Imagery and Mapping Agency (NIMA) is pursuing integrated process improvement using the FAA-iCMM. The FAA has offered FAA-iCMM training to external organizations on numerous occasions.

Summary and Conclusions

The FAA replaced three separate CMMs with the FAA-iCMM, containing all features of its source CMMs, and is successfully applying the FAA-iCMM to achieve integrated process improvement. Take a look at your organization, and consider whether integrated process improvement makes sense. If you find it does, then, like the FAA, software process improvement may not be enough for you either.

References

[1] Paulk, Mark et. al. Capability Maturity Model for Software, Version 1.1, February 1993, CMU/SEI-93-TR-24 and CMU/SEI-93-TR-25, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.

[2] Bate, Roger et. al., A Systems Engineering Capability Maturity Model, Version 1.1, November 1995, SECMM-95-01, CMU/SEI-95-MM-003, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.

[3] Ferguson, Jack et. al., Software Acquisition Capability Maturity Model (SA-CMM), Version 1.01, December 1996, CMU/SEI-96-TR-20, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.

[4] Ibrahim, Linda, et. al., The Federal Aviation Administration Integrated Capability Maturity Model® (FAA-iCMM), Version 1.0, Federal Aviation Administration, November 1997. (available at www.faa.gov/aio)

[5] Ibrahim, Linda, et. al., The Federal Aviation Administration Integrated Capability Maturity Model Appraisal Method, Version 1.0, Federal Aviation Administration, April 1999. (available at www.faa.gov/aio)

[6] Ibrahim, Linda, "Using an Integrated Capability Maturity Model - The FAA Experience", Proceedings of the Tenth Annual International Symposium of the International Council on Systems Engineering (INCOSE), 16-20 July 2000, Minneapolis, MN, pp. 643-648.

[7] Ibrahim, Linda, "Smart Buying with the Federal Aviation Administration's Integrated Capability Maturity Model", CROSSTALK, November 1998.

Additional Resources:

The FAA Process Improvement (FAA-iCMM) Web page. (www.faa.gov/aio/ProcessEngr/iCMM/index.htm) contains further details on FAA iCMM version 1.0 including the FAA iCMM Appraisal Method version 1.0 in PF Wros and HTML.

Biography

Dr. Linda Ibrahim joined the FAA in 1996. She is the FAA Chief Engineer for Process Improvement. She formulated the concept and led the development and implementation of the FAA's integrated Capability Maturity Model and its appraisal method. Linda received the Meritorious Achievement Award of the US Secretary of Transportation for her innovative work in CMM integration.

Linda has been working in software engineering for over 30 years, as practitioner, educator, and researcher; in government, industry, academic, and applied research environments; in the US, Europe, and the Middle East. She is a member of the Steering Group for the DoD/NDIA sponsored CMM Integration (CMMI) project.

Linda holds a BA in Mathematics, MS in Information Science, and Ph.D. in Electrical Engineering. She is a member of INCOSE, ACM, and IEEE.

Contact Information

Dr. Linda Ibrahim
Chief Engineer for Process Improvement
Federal Aviation Administration
800 Independence Ave, SW
Washington, DC 2059

Phone: (202) 267-9878
E-Mail: linda.ibrahim@faa.gov
URL: www.faa.gov

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

Toward Defining A Software Engineering Body of Knowledge

Abstract

The focus of this paper is to report on our initial work in defining and categorizing a Software Engineering Body of Knowledge (SwE-BOK). This project was sponsored by the FAA to build a software competency model for their research and acquisition work force. Since there was no defined software engineering body of knowledge in the literature upon which to build such a model, we embarked upon the task of developing one. It is our hope that the SwE-BOK will also be usable by other competency modelers and curriculum designers.

This paper offers a view of the content, categorization, and organization of the software engineering domain that is not currently reflected in the literature. It is presented as a hierarchical structure organized into knowledge categories, knowledge areas, and knowledge units. We also offer examples of its application to competency model building and to education and training curriculum design.

Introduction

Software plays an increasingly important and central role in all aspects of daily life: in government, banking and finance, education, transportation, entertainment, medicine, agriculture, and law, to name only a few. The number, size, and application domains of software programs being developed have grown dramatically. As a result, billions of dollars are being spent on software development, and the livelihood and lives of millions directly depend on the effectiveness of this development. Unfortunately, there are severe problems in the cost, timeliness, and quality of many software products. Particularly serious is the effect these problems can have on the safety-critical elements of software, which are central to many real-time embedded systems. These problems are attributed to the short history of software and immaturity of its discipline.

In recent years, there have been a number of studies and commentaries on the immaturity of the software engineering profession [9,14]. In 1996, a study of the profession by Ford and Gibbs [8] designated eight infrastructure components that could be used to evaluate a mature profession: initial professional education, accreditation, skill development, certification, licensing, professional development, a code of ethics, and a professional society. They went on to classify these components using four levels of maturity. At that time, they argued that the only component at a relatively mature level was Professional Development. If one accepts the argument that the profession is not mature, then a number of questions arise: At what stage are we in the evolution of software development as an engineering discipline? What is (or should be) the content of the profession – what is the body of knowledge? How should software engineers be prepared? By which means do we recognize or assure professional work?

Since 1993, the IEEE-CS and ACM have been actively promoting software engineering as a profession. Special task forces have been directed at establishing a software engineering code of ethics and professional practice [11], determining accreditation criteria for software engineering programs [4] supporting curriculum development, and preparing a guide to the software engineering body of knowledge [6]. In 1998, the IEEE-CS and the ACM formed the Software Engineering Coordinating Committee (SWECC) to foster the evolution of software engineering as a professional computing discipline and to coordinate the various software engineering "maturing" activities. In addition, there have been a number of other activities that support defining the profession: the State of Texas has begun to develop a process for the licensing of professional engineers in software engineering [3]; the Working Group on Software Engineering and Training is defining a set of guidelines that support the development of a software engineering curricula; and Lethridge [13] has completed a study and analysis of the knowledge needed and used by practicing software engineers. While the work by the IEEE-CS/ACM holds great promise for defining the software engineering body of knowledge in depth, the completion of the project is still far in the future. All of their activities, however, support and motivate the need for an organized and comprehensive definition and categorization of software engineering knowledge.

This paper reports on our attempt to define a software engineering body of knowledge. This work is part of a larger project sponsored by the Federal Aviation Administration to

by Richard Turner, OUSD Software Intensive Systems, Dr. Iraj Hirmanpour, Dr. Soheil Khajenoori, and Dr. Thomas B. Hilburn of Embry-Riddle Aeronautical University

develop a software engineering competency model aimed toward assessing the capability of their research and acquisition work force.

Background

In 1998, the FAA initiated a project designed to help improve the software engineering knowledge and abilities of their Office of Research and Acquisitions (ARA) personnel. The project involved the development of a software competency model for the FAA documented roles.

In order to build the model it was necessary to define and structure a SwE-BOK.

In order to get a firm grip on the problem and to keep the scope within available time and resources, we decided in advance to ignore the role of system engineering within software engineering. We also excluded any areas of knowledge that may be supportive of software engineering but do not play a direct role in software development. For example, discipline areas such as continuous mathematics, the natural sciences, traditional engineering science, psychology, economics, and business administration were not included the SwE-BOK to keep the scope within our objective.

We determined that the SwE-BOK primary purpose would be to:

- assist with the design of software engineering competency models
- aid in software engineering curriculum design,
- help develop skill assessment programs and tools,
- help with the development of competency based training programs.

Knowledge Architecture

In order to achieve a balance between simplicity and clarity, and the appropriate depth and detail of knowledge description, we chose three levels of abstraction to partition the software engineering body of knowledge. The three layers are named Knowledge Category (KC), Knowledge Areas (KA), and Knowledge Units (KU). We assumed that the universe of SwE-BOK can be divided into four major categories that we have named Computing Fundamentals, Software Product Engineering, Software Management, and Software Domains. Knowledge categories are; therefore, subdisciplines of software engineering that are generally recognized as representing a significant part of the body of software engineering knowledge. In our classification scheme, knowledge categories are high-level structural elements used for organizing, classifying, and describing software engineering knowledge.

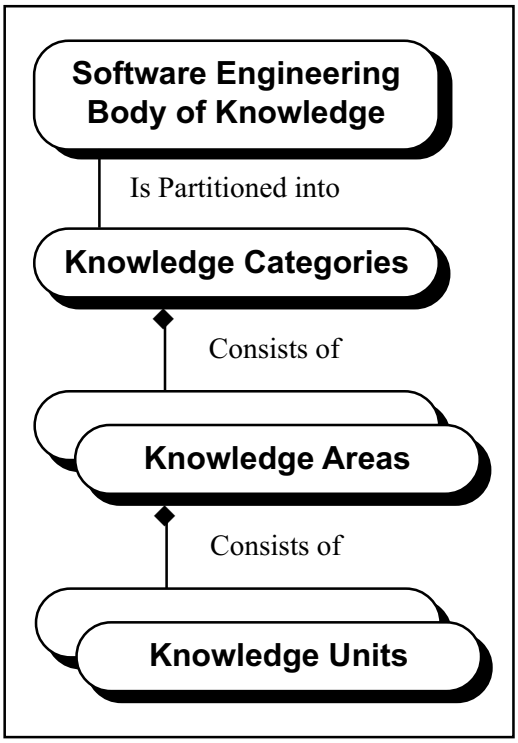


Figure 1: SwE-BoK Architecture

Each Knowledge Category is then divided into a group of Knowledge Areas. A Knowledge Area is a subdivision of a KC that represents software knowledge that is logically cohesive and related to the KC through inheritance or aggregation. Each Knowledge Area is then decomposed into a set of Knowledge Units (KU). KUs are subdivision of a KA that represents a basic component of SwE knowledge with a crisp and explicit description. For the purposes of this activity, the KU is "atomic" (i.e., will not be subdivided into simpler or more basic elements). See Figure 1.

In summary, an aggregation of Knowledge Units creates a Knowledge Area, and similarly, an aggregation of Knowledge Areas creates a Knowledge Category. Knowledge Units are dependent on the current state of technology and change over time while KA and KC are invariant over time. For example, one of the Knowledge Units specified in the BOK is Architectural Design. The contents of such a unit will change as knowledge about architectural design increases. However, the Knowledge Area of software design and knowledge Category of Product Engineering will stay invariant.

Continued on page 10

Continued from page 9

For the purposes of this work, the term "knowledge" is used to describe the whole spectrum of content for the software engineering discipline: information, terminology, artifacts, data, roles, methods, models, procedures, techniques, practices, processes and literature. In our model, we organized software engineering knowledge into three Knowledge Categories: Computing Fundamentals, Software Product Engineering, and Software Management. Since the application domain of software has its own specific knowledge requirements, we created a fourth category called Software Domains. Although this decomposition and the corresponding descriptions encompass a significant portion of the knowledge considered part of the software engineering profession, there are particular areas related to special domains or areas on the periphery of software engineering that are not included – such as GUI and human factors. Some areas in this SwE-BOK have also been given more attention and focus because of the context of the project. An example is the inclusion of a description of a "software acquisition" Knowledge Area in the Software Management category.

In the following sections, each of the Knowledge Categories is described, and the Knowledge Areas that are part of each KC are detailed. It should be evident that the body of knowledge has varying levels of decomposition and exposition, depending upon the current state of knowledge in a particular component and the significance of the component to the overall description of the software engineering discipline.

Computing Fundamentals Category

Computing Fundamentals include the knowledge, concepts, theory, and skills of computing that form the foundation for the development of software and the discipline of software engineering. This category is divided into the following Knowledge Areas:

1. Algorithms and Data Structures
2. Computer Architecture
3. Mathematical Foundations
4. Operating Systems
5. Programming Languages

Software Product Engineering Category

Software Product Engineering is concerned with a well-defined and integrated set of activities to produce correct, consistent software products effectively and efficiently. Software Product Engineering includes the technical activities of producing a software product such as requirements engineering, design, coding and testing. These engineering activities involve documenting software work products and maintaining trace ability and consistency between them. This category includes knowledge about the controlled transition between the stages of the software life cycle and the activities needed to deliver quality software products to the customer. This category is divided in to the following Knowledge Areas:

1. Software Requirements Engineering
2. Software Design
3. Software Coding
4. Software Testing
5. Software Operation and Maintenance

Software Management Category

Software Management deals with the concepts, methods and techniques for managing the development or acquisition of software products. Software management includes activities concerned with project management, with management of risk, with the configuration of a software system, and with knowledge about how to produce high-quality software. This category is divided into the following Knowledge Areas:

1. Software Project Management
2. Software Risk Management
3. Software Quality Management
4. Software Configuration Management
5. Software Process Management
6. Software Acquisition

Software Domains Category

The Software Domains category concerns knowledge about specific domains that involve the application and utilization of software engineering knowledge from the other categories. In this context, a software domain represents an area of knowledge that has two features: its technical content is required in the development of a particular type of software; and it represents a significant field of knowledge, with an established organization and content, that is documented by

research and application. This category is divided into the following Knowledge Areas:

1 Artificial Intelligence
2 Database Systems
3 Human-Computer Interaction
4 Numerical and Symbolic Computing
5 Computer Simulation
6 Real-Time Systems

The four categories of SwE-BOK are thus decomposed into the 22 Knowledge Areas listed. These areas are then further subdivided into 64 Knowledge Units. For brevity, we do not list all the 64 Knowledge Units. However, for a more detailed description of Knowledge Areas and Knowledge Units the reader is referred to http://computing.db.erau.edu/SEERS which contains the latest work in this area.

Applying the BOK

The SwE-BOK can be used for a variety of purposes: it can help to solve problems and address issues in industrial and academic settings. Most importantly, it can serve as a general model for understanding and describing the software engineering profession. Such an understanding is essential to the maturing of a discipline and is a necessary step in determining the professional standards and procedures required for the effective practice of software engineering. In this spirit, the SwE-BOK could be used to develop criteria and assessment instruments for the certification and licensing of software engineers. In addition, the SwE-BOK could be used by individual engineers to assess their own knowledge about the software engineering profession and provide a framework for them to plan for their professional development.

In this section we discuss two ways that the SwE-BOK could be applied: to help develop a competency model for an industrial software organization and to design a software engineering curricula for an academic environment.

Software Engineering Competency Model

In an industrial setting, the SwE-BOK can help organizations to examine, categorize, and organize their software activities and identify the knowledge levels required to carry out their specific software tasks. Organizations that want to assess the knowledge of their engineers could, for example, use the SwE-BOK in designing a competency evaluation system. Such a competency system could use the SwE-BOK to help identify and judge what kind of software engineering knowledge is required to accomplish the tasks associated with individual software-related roles. An assessment of this nature might also be coupled by using the SwE-BOK to design training programs and develop an overall strategy to improve an organization's software capabilities.

A competency model should provide a framework for identifying the required competencies of a successful work force. A competency model was developed as a part of the FAA project. The competency model identifies software roles, software activities related to the role, and maps the role's activities according to the knowledge components, from the SwE-BOK, that are required to carry out the activities.

Software Engineering Curriculum

In an academic setting, the SwE-BOK can provide faculty with the basic information about software engineering knowledge, to support the development of educational curricula in software engineering. In addition, the SwE-BOK can support the creation of a set of more general guidelines for the development and accreditation of software engineering programs by groups such as ABET (Accreditation Board for Engineering and Technology) and CSAB (Computer Science Accreditation Board). The SwE-BOK can provide the foundation for such development and can serve as a framework for curriculum design.

The Knowledge Categories and Knowledge Areas could serve as high-level curriculum components, while the Knowledge Units could serve as the basis for detailed module and course design. Table 1 illustrates how the SwE-BOK can be used to guide and structure a high-level curriculum design for an undergraduate program in software engineering. (In this example we are assuming that the objective of the curriculum is to produce software engineers prepared to develop software for real-time, embedded systems.)

Each of the SwE-BOK Knowledge Categories and Knowledge Areas are listed in Column 1 of the table (for completeness, non-software components are also listed). Column 2 indicates the depth of knowledge required for each of the components (we use three knowledge depth levels: Awareness,

Continued on page 12

Continued from page 11

Understanding, and Execution). Inclusion of the metric compels curriculum designers to be more careful and thoughtful about design decisions. Column 3 provides semester hour credits associated with the various components; this provides high-level constraints on the future course and module development.

Table 1: An Example Architecture for a Software Engineering Curriculum

Knowledge Components	*Depth of Knowledge*	*Credit Hours*
Computing Fundamentals	**Category**	24
Algorithms and Data Structures	Execution	
Computer Architecture	Execution	
Mathematical Foundations	Understanding	
Operating Systems	Execution	
Programming Languages	Understanding	
Software Product Engineering Category		15
Software Requirements Engineering	Understanding	
Software Design	Execution	
Software Coding	Execution	
Software Testing	Execution	
Software Operation and Maintenance	Understanding	
Software Management Category		9
Software Project Management	Execution	
Software Risk Management	Understanding	
Software Quality Management	Execution	
Software Configuration Management	Understanding	
Software Process Management	Understanding	
Software Acquisition	Awareness	
Software Domains Category		9
Artificial Intelligence	Awareness	
Database Systems	Understanding	
Human-Computer Interaction	Understanding	
Numerical and Symbolic Computing	Awareness	
Computer Simulation	Understanding	
Real-Time Systems	Execution	
General Education	Understanding	51
Mathematics		12 Credits
Science		12 Credits
Communication		12 Credits
Humanities and Social Sciences		15 Credits
Application Domain	**Understanding**	12 Credits
		Total = 120 Credits

Unfortunately, the current state of software engineering experience and knowledge in academia is in short supply. The curriculum example in Table 1 illustrates how the SwE-BOK can be helpful, especially for faculty without a strong software engineering background, for the design and development of a software engineering curriculum.

Conclusions

There is a need for a defined software engineering body of knowledge, in particular, to provide the basis for the following activities:

- design of competency and skill assessment programs and tools,
- development of competency based training programs, and
- software engineering curriculum design.

The structure and organization of the body of knowledge described in this paper allows for both high level and detailed level design of software engineering curriculum. The knowledge categories and knowledge areas can serve as a guideline to structure the curriculum and knowledge units can provide the necessary detail to design the course contents. We believe base-lining and cataloguing the current

body of knowledge will help advance the state of software engineering both as a discipline and as a profession.

References

[1] ACM Task Force on the Core of Computer Science, Computing as a Discipline. Communications of the ACM 32, (January 1989), 1-5.

[2] The Full Computing Classification System, 1998 Version. Computing Reviews (January 1998), Association for Computing Machinery.

[3] Bagert, D. Taking the Lead in Licensing Software Engineers. Communications of the ACM 43, 4, (April 1999), 27-29.

[4] Barnes, et. al. Draft Software Engineering Accreditation Criteria. Computer 31, 4 (May 1998), 73-75.

[5] Bloom, B. S. (Ed.) The Taxonomy of Educational Objectives: Classification of Educational Goals. Handbook 1: The Cognitive Domain. New York: McKay Press, 1956.

[6] Bourque, P., et. al. Guide to the Software Engineering Body of Knowledge, A Straw Man Version. IEEE Computer Society, September 1998.

[7] Davis, G. B., et. al. IS'97: Model Curriculum for Undergraduate Degree Programs in Information Systems, 1997, On-line at www.acm.org/edcuation/curricula.html.

[8] Ford, G. and Gibbs, N.E. Mature Profession of Software Engineering, CMU/SEI-96-TR-004. Software Engineering Institute, Carnegie Mellon University, 1996.

[9] Gibbs, W.W. Software's Chronic Crisis. Scientific American 271, 3 (September 1994), 86-95.

[10] Hilburn T., Hirmanpour, I., Khajenoori, S., Turner, R. and Qasem, A. A Software Engineering Body of Knowledge, Version 1.0. Technical Report CMU/SEI-99-TR-004, Software Engineering Institute, Carnegie Mellon University, April 1999.

[11] IEEE-CS/ACM Joint Task Force on Software Engineering Ethics and Professional Practice. Software Engineering Code of Ethics, On-line at www.computer.org/tab/seprof/code.htm

[12] Khajenoori, S., Hilburn T., Hirmanpour, I., Turner, R. Qasem, A. Software Engineering Competency Study Final Report, ERAU-FAA Project, December 1998.

[13] Lethbridge, T.C. The Relevance of Software Engineering Education: A Survey and Some Recommendations. Annals of Software Engineering 6, (1998), 91-110.

[14] Shaw, M. Prospects for an Engineering Discipline of Software. IEEE Software 7, 6 (November 1990) 15-24.

[15] Tucker, A. B. (Ed.) Computing Curricula 1991: Report of the ACM/IEEE-CS Joint Curriculum Task Force. IEEE Computer Society Press, 1991.

[16] Ibrahim, L. et. al. The Federal Administration Integrated Capability Maturity Model, Version 1.0, Washington DC; Federal Aviation Administration, November 1997.

Biographies

Dr. Iraj Hirmanpour has over 30 years leadership experience in education and industry consulting, an established record as a strategic thinker, a team-based manager, and a people motivator. He successfully re-engineered three different departments over the course of his academic career. As an academic researcher working with industry he lead development of the Operations Maturity Model (OMM) for Andersen Consulting, development of the Software Engineering Body of Knowledge (SoW-BOK) for FAA published as an Software Engineering Institute (SEI) Technical Report, and the development of a Software Engineering Competency Model published as an FAA Technical Report. He has also worked on the Process Improvement Body of Knowledge published as an SEI Technical Report. Research and consulting is in process improvement using improvement models such as the Capability Maturity Model (CMM), Personal Software Process (PSP), and Team Software Process (TSP) from the SEI. He has published and presented extensively on software engineering and information technology subjects.

Dr. Thomas B. Hilburn received his Ph.D. in Mathematics from Louisiana Tech University in 1973. Prior to that he served in the US Navy where he taught courses in computer programming, and inertial and satellite navigation; he completed an electronics program at the Navy Postgraduate School in Monterey California; and he served a tour in Vietnam as the Officer-In-Charge of the APL-5 . He joined the mathematics faculty at Embry-Riddle Aeronautical University in 1973 and since 1989 has served as a Professor of Computer Science. His current interests include software engineering, formal methods, individual and team software processes, and computer science and software engineering education. He has worked with the MITRE Corporation on a problem involving the analysis of airspace complexity and has presented special short courses on Ada and software engineering at General Electric-Daytona Beach, Martin Marrietta-Daytona Beach, and the National Civil Air Training Organization in Cairo, Egypt. In 1996-1997 he spent a year as a visiting scientist at the Software Engineering Institute, Carnegie Mellon University.

Continued on page 14

Software Engineering Body of Knowledge

Continued from page 13

Dr. Soheil Khajenoori is the Director of Master of Software Engineering Program at Embry-Riddle Aeronautical University. He worked at GPT Stromberg-Carlson, a major telecommunications company, for the development of software engineering competency. His responsibilities included development of a software engineering training program, integrating Computer Aided Software Engineering (CASE) into the product development life cycle, and participating in software development activities (e.g., analysis, design, evaluation) as a consultant to the projects. He has been working with several organizations such as Motorola Paging Products Group and Boeing Space Division/KSC, on the Personal Software Process paradigm, and individual and small teams software process improvement initiatives. In addition, he has been a consultant to major companies on software engineering subjects such as training, and technology transfer issues.

Dr. Khajenoori has been Project Lead on fifteen research projects and has published a number of articles on software engineering subjects including CASE, software development methodologies (Structured and Object-Oriented methods), software metrics, and software process engineering and improvements. He is currently teaching courses on Personal Software Process, Software Requirements Engineering and Software Design, as a part of the MSE program. For the past three years he has been closely working with Software Engineering Institute (SEI), Watts Humphrey and Iraj Hirmanpour on implementation of Personal Software Process (PSP) and Team Software Process Paradigms.

Contact Information

Dr. Iraj Hirmanpour, Department Chair
Dept. of Computing & Mathematics
Embry-Riddle Aeronautical Univ.
600 Clyde Morris Boulevard
Daytona Beach, FL 32114

Phone: (904) 226-6691
Fax: (904) 226-6678

E-Mail: Ih@db.erau.edu
URL: http://faculty.db.erau.edu/iraj/

Dr. Thomas Hilburn
Dept. of Computing & Mathematics
Embry-Riddle Aeronautical Univ.
600 Clyde Morris Boulevard
Daytona Beach, FL 32114

Phone: (904) 226-6889
Fax: (904) 226-6678

E-Mail: Hilburn@db.erau.edu
URL: http://faculty.db.erau.edu/hilburn/

Dr. Soheil Khajenoori
Masters of Software Engineering
Embry-Riddle Aeronautical Univ.
600 Clyde Morris Boulevard
Daytona Beach, FL 32114

Phone: (904) 226-7036
Fax: (904) 226-6678

E-Mail: Soheil@db.erau.edu
URL: http://computing.db.erau.edu/ourteam/home_pages/soheil/

Richard Turner
OUSD(AT&L)/ARA/SIS
1931 Jefferson Davis Highway
Crystal Mall-3, Suite 104
Arlington, VA 22202

Phone: (703) 602-0851-x124
Fax: (703) 602-3560

E-Mail: rich.turner@osd.mil

DoD Software Tech News Editorial Board

Lon R. Dean, Editor, DoD Software Tech News
ITT Industries

Paul Engelhart, DACS COTR
Air Force Research Laboratory Information Directorate/IFTD

Elaine Fedchak
ITT Industries

Dr. Jack Ferguson
ODUSD(S&T)

Morton A. Hirschberg, Editorial Board Chairman
US Army Research Laboratory (Retired)

Phil King
ITT Industries

Thomas McGibbon, DACS Director
ITT Industries

Dave Nicholls, DACS Deputy Director
ITT Industries

Marshall Potter
Federal Aviation Administration

Nancy L. Sunderhaft
ITT Industries

The *DoD Software Tech News* is published quarterly by the Data & Analysis Center for Software (DACS). The DACS is a DOD sponsored Information Analysis Center (IAC), operated by ITT Industries, Advanced Engineering and Sciences Division, and administratively managed by the Defense Technical Information Center (DTIC).

Application of Software Assurance Best Practices to Communications, Navigation, Surveillance (CNS) and Air Traffic Management (ATM) Systems

by Ronald L. Stroup, FAA

Introduction

The Federal Aviation Administration (FAA) is charged with maintaining a safe air transportation system within the United States, and in Oceanic Flight Information Regions assigned to the U.S. by the International Civil Aviation Organization (ICAO). This system involves aircraft and a supporting infrastructure of procedures, personnel, and equipment. Software is an essential element of the National Airspace System (NAS) including the aircraft, and embodies much of its functionality. Developed and acquired by the FAA as a portion of a system or system component, software can be a major cost and schedule risk factor in the procurement of new systems and the modernization of existing systems.

Current design practices for navigation systems have employed real-time embedded systems design techniques that rely on good software engineering practices to achieve the extremely high integrity and continuity of safety requirements. A number of guidelines have been developed to streamline the procurement of software-based systems but most guidelines do not address the assurance of software in the context of safety critical systems and environments. Nor do they address the assurance of performance with sufficient rigor for operational deployment. While guidelines for Software Development Assurance have been available for aircraft through the use of RTCA/DO-178B/ED-12B, *Software Considerations in Airborne Systems and Equipment Certification*, no corresponding guidelines have been universally in place for ground systems[1].

Until recently, the evaluation of ground systems was done separately from aircraft systems. However, current NAS architectural designs rely on coupled systems that require integrated analysis and mitigation of potential safety hazards. This leads to a coordinated allocation of safety objectives between airborne and ground systems. Without a consistent and equivalent set of guidelines between ground and aircraft systems, the FAA will have an extremely difficult task in assuring the safety of modern systems.

The Proposed Solution

The FAA's *Guidelines for the Application of Software Assurance Best Practices to Communications, Navigation, Surveillance (CNS) and Air Traffic Management (ATM) Systems* is to ensure consistency and an acceptable level of confidence in the development of the software components of the NAS. Techniques available through the system safety assessment process result in risk mitigation strategies that minimize the impact of software design errors. This mitigation affects a relatively small portion of the overall systems software that can adversely affect safety of flight. The level of effort required in implementing this guidance will vary as a function of the operational use and criticality of function of that software.

Consistent development assurance practices will help to achieve technical goals for the safety requirements, interoperability, performance and security assurance, as well as management control, while promoting responsible expenditure of public funds.

A Common Sense Approach

These guidelines provide common sense approaches to ensuring new developments and changes to existing systems meet the NAS operational safety requirements. More important, every effort has been made to take advantage of architectural approaches which significantly reduce the amount of software that must be assured in accordance with these guidelines.

A great deal of effort has already been invested in process improvement (FAA-iCMM®) and the Acquisition Management System (AMS) for software based systems developed and procured by the FAA. These guidelines were developed to take advantage of this work. These guidelines do not replace software development processes or software acquisition processes. These guidelines supplement those processes and provide a comprehensive set of assurance activities throughout the software development life cycle.

They were also based on the experience and knowledge gained through:

- Successful review and approval of the Special Category - I (SCAT-I) Differential Global Positioning System (DGPS)
- Approval of the Transponder Landing System (TLS)

Continued on page 16

1. Throughout this document, the term "Ground systems" is used to refer to all non-airborne systems, that is, systems whose physical location is other than aboard an aircraft. Specifically, "non-airborne" systems as used here comprises both Ground-based systems and the more-recent satellite-based systems; however, we continue to use the term "Ground-based" both for historical reasons and to avoid such awkward expressions as "non-airborne systems" and "Ground-based and satellite-based systems."

Continued from page 15

- Ongoing efforts in the Wide Area Augmentation System (WAAS)
- Ongoing work in RTCA Special Committees SC-189 working on global implementation of CNS/ATM modern concepts, and SC-190 working on clarifying the guidance in DO-178B
- FAA Streamlining Software Aspects of Certification (SSAC) efforts
- Applying DO-178B on SCAT-1, WAAS, and TLS has demonstrated a positive effect on achieving the performance standards for these systems

The major impact of the guidelines to affected systems will be in the area of verification. These guidelines have very specific rigorous requirements for verification that are not provided by previous ground systems guidance materials.

Contents of the Guidelines

The guidelines contain an overview of the end-to-end system safety process and how it is used to establish the risk control required for software. A principal purpose of this overview is to demonstrate the use of architecture in reducing the amount of software that needs to be assured. Conceptually, the guidelines contain the lessons learned in software development assurance. Experts within the realms of system safety, airborne software development assurance, ground systems acquisition and maintenance both from the FAA and industry are contributing to the document.

An approval process is presented which provides a framework for assuring the adequate evaluation of software-based systems. This process recognizes the existing approach to certification and commissioning of systems. The process also addresses the challenges that will be encountered by future systems. Detailed guidance on adapting software development assurance best practices to the ground systems environment is provided, stressing the commonality to existing guidance. The guidance demonstrates how the software development assurance process fits into the existing approval, evaluation, and procurement processes of the FAA. Specific attention is devoted to Commercial-Off-The-Shelf software, Legacy Systems, and the introduction of new technology. The guidance is linked to the appropriate hazard analysis process to ensure that risk is controlled.

The author is currently participating in RTCA Special Committee 190/EUROCAE Working Group 52 to develop associated guidance for ground systems. While the RTCA document, by necessity, will be generic, the FAA guidance will provide those details that will assist the user in implementation of the RTCA guidance. Every effort has been made to ensure that the resulting guidance contained in the FAA document remains consistent with the output of RTCA Special Committee 190.

Summary

The safety of flight operations is dependent on more than the design of the aircraft and its associated equipment. Equally important to safety are ground and flight personnel procedures, ground-based equipment, and space-based equipment. Various techniques or mitigation strategies are used to minimize the risk and impact of equipment design hazards. With proper architecture and other mitigation approaches, any portion of the remaining potential residual errors will be attributed to software. However, this results in a relatively small portion of the overall systems software that can adversely affect safety of flight. When software design errors can potentially affect safety of flight, a means is needed to establish assurance or confidence that there is a sufficient level of dependability for this software. RTCA DO-178B, Software Considerations in Airborne Systems and Equipment Certification provides an accepted approach for establishing assurance/confidence in the dependability of safety critical airborne applications. To provide a consistent, defensible approach, any systems that can adversely affect safety of flight should adopt equivalent software development assurance best practices as a basis for establishing and achieving the required level of assurance of critical systems. These guidelines provide the minimum guidance necessary to apply software development assurance best practices specifically to ground systems applications. Our goal is to do this efficiently with minimum disruption and cost to fielded products, ongoing development, and future developments that have consequences for the safety of flight operations.

Biography

Ronald L. Stroup joined the Federal Aviation Administration as an Aerospace Engineer in 1989. He holds a Bachelor of Science in Avionics Engineering from Parks College of Saint Louis University. Mr. Stroup served as a systems engineer in the Aircraft Certification Services' Chicago Aircraft Certification Office and in 1997 became the Software Technology Specialist for the Aircraft Certification Service. His

responsibilities included providing technical expertise in the area of software approvals and acted as a focal point to improve the software approval process.

Since 1998, Mr. Stroup has served as the Software Safety and Certification Lead for the Office of Information Services and Chief Information Officer. His duties include developing and applying software assurance standards to the acquisitions of software-intensive National Airspace Systems.

Mr. Stroup is a member of the RTCA/SC-190 Committee, IEEE, and Co-Program Manager for the Streamlining Software Aspects of Certification initiative. He served as a Subject Matter Expert for the Software Fundamentals and Software Procedures course to develop training in the training in the application of RYCA/DO-178B Assurance Standard and Software Engineering Practices. Mr. Stroup serves on the FAA's Systems Engineering Council and the System Safety Working Group.

Contact Information

For further information about this article please contact the author at

Ronald L. Stroup
Office of Information Services
Process Engineering Division
Federal Aviation Administration
800 Independence Ave. SW
Washington, DC 20591

Phone: (202) 493-4390

E-Mail: Ronald.L.Stroup@faa.gov

On November 1, 1958, Elwood R. Quesada took the oath as FAA's first Administrator

The current Administrator is Jane F. Garvey. She was sworn in August 4, 1997 as the 14th Administrator of the Federal Aviation Administration. She is the first Administrator confirmed by the Senate to a five-year term.
www.faa.gov/jgarvey/welcome.htm

Building Trustworthy Systems: An FAA Perspective

The security and economic viability of our nation depend on an effective and uncompromised transportation infrastructure, and information systems security (ISS) is at the heart of keeping the air transportation infrastructure protected. In an earlier paper, we introduced the Federal Aviation Administration's (FAA's) layered approach to ISS [1]. We described in that paper the rationale and causes for a new emphasis on ISS at the FAA and the holistic and multidimensional approach that we are taking to address this challenge. For the FAA, ISS has a more expansive meaning than it had in the traditional computer security environment. The most important results that we focus on at the FAA are the safety, security, and efficiency of the National Airspace System. We demonstrated in our earlier paper that the structural, process, and operational foundations of ISS provide the vital mechanisms necessary for the FAA to build, certify, operate, and maintain trustworthy systems in support of achieving those results. In the earlier paper, we concentrated on describing in detail the structural portion of our model which is illustrated in the form of a pyramid (Figure 1) that includes five layers of protection: personnel security, physical security, compartmentalization and information systems security, site specific adaptation, and redundancy. In this paper, we will describe the other two key elements of our approach—the process and operational models—and we will show how they interrelate with the ISS pyramid to establish a balanced overall information security program. In addition, we will present an evolution of the ISS pyramid structure into a more generic cyber security model which will be helpful for prioritization analyses and for architecture and engineering efforts to guide the long-term ISS Program.

Background

For our purposes, we define a trustworthy system as one that does what we expect it to do and not something else, despite environmental disruptions, human and system errors, and attacks from hostile parties. Design and implementation errors must be avoided and/or eliminated, or the system must be built with a degree of robustness such that potential problems are either tolerated or have no operational impact. It is not sufficient to address only some of these dimensions, nor is it sufficient to simply assemble components that are themselves trustworthy. As a consequence, trustworthiness is holistic and multidimensional [2].

The FAA's mission is to ensure safe, secure, and efficient air travel in the National Airspace System. FAA's ability to fulfill this mission depends on the adequacy and reliability of its air traffic control (ATC) system. Faced with rapidly growing air traffic volumes and aging air traffic equipment, the FAA initiated an ambitious ATC modernization program. This program includes the acquisition of a vast network of radars, computers, navigation and communications equipment, in addition to new facilities and support equipment. It also includes deployment of FAA information systems, which are very large and complex, and are increasingly integrated, compounding the risks to the trustworthiness of the system in its entirety. These systems process a very wide range of different types of information including radar, weather, flight plans, surveillance, navigation/landing guidance, traffic management, voice, network management, aircraft certification, inspection, flight standards, regulatory information, and runway status. This information is normally available to be read by a very large user base and is for the most part not considered "confidential" with regards to "read access." These resources reside at, or are associated with, several types of ATC facilities—air traffic control towers, terminal radar approach control (TRACON) facilities, air route traffic control centers (enroute centers), flight service stations, and air traffic control system command centers (ATCSCC). The challenge for the FAA, then, is to ensure that this vast and complex information centric network, which includes a diverse array of data sources, users, and facilities, remains trustworthy, uncompromised and ready for operation on a 365 x 24 basis.

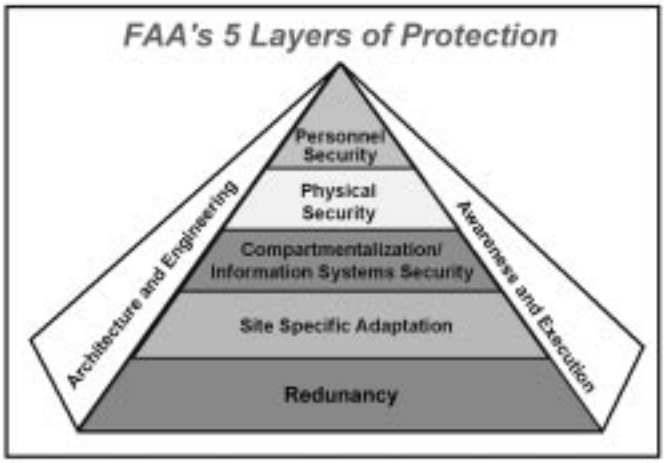


Figure 1: Layers of Protection

The Process Model

While the Structural Model of Figure 1 provides a good understanding of key conceptual and systemic actions that need to be taken to implement an effective ISS Program, it does not define how we are going to do it. To describe the "how-to" level, we looked at the problem from an additional

perspective and developed a process model to guide our ISS Program implementation. FAA systems, in particular National Airspace subsystems, are generally accessible and physically located in many facilities across the nation. To establish that a facility has an operational capability to protect itself, three threads—system, personnel, and physical—depicted in the Process Model of Figure 2 need to be addressed. It is not enough to ensure that individual systems are protected; we must also ensure that each facility as a whole is protected with regard to system, physical, and personnel security.

Since the systems thread is perhaps the newest challenge to the FAA, we will deal with that thread first. The comprehensive system certification process used by the FAA is represented in Figure 3. At the beginning of this process, we conduct a risk and vulnerability assessment. Without a thorough knowledge and understanding of the risks and vulnerabilities, it is not possible to develop adequate protection mechanisms. After this is accomplished, a System Certification and Authentication Package (SCAP) is developed which includes such items as a risk mitigation plan, an ISS plan, an ISS test plan, protection profiles that define the security requirements of the system, and a certification statement. The system developer or owner must provide evidence that the system is ready for certification. Then the Chief Information Officer reviews and provides the certification of the system, after which it can be deployed with the concurrence of the Designated Approving Authority (DAA) of the business unit that is deploying the system.

Beyond the "national" certification of systems described above, there is also a need to ensure that facility system protection, which applies to the local, site-specific configuration of nationally distributed systems, to local adaptations, and to systems that are under the purview of the facility, is also in place. The primary steps to assure that these systems are secure are taken by local facility personnel.

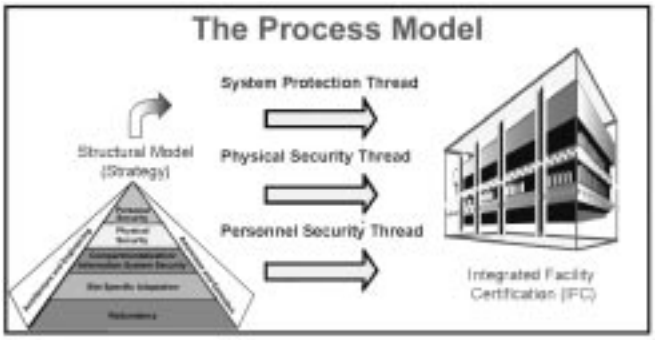


Figure 2: The Process Model

The approach envisioned by the FAA to protect its systems generally follows the following phases.

- Assessment: The system is identified and its vulnerabilities are analyzed through penetration testing and vulnerability analysis.
- Planning: Risks are analyzed and prioritized and decisions are made on how to address them. Where the decision is made to address a risk by modifications to the system, a plan is developed to remediate the system.
- Remediation: Vulnerabilities are addressed by adding new or modifying existing features and a test plan is developed to prove that vulnerabilities have been addressed.
- Certification and Authorization: The test plan is executed and, if the outcome is positive, the system is certified. The DAA determines, based on the available documentation, whether or not to authorize for operation or continue operation of the system.
- Deployment and Commissioning: Systems that have undergone remediation activities and have been certified and authorized are deployed for operation.

While the system thread which we have discussed thus far is very important, the personnel and physical threads are no less vital. To protect the National Airspace System, each facility is designed and constructed with personnel and physical security barriers. These personnel and physical barriers include security fences, cipher locks, background checks, security clearances, identification badges, and

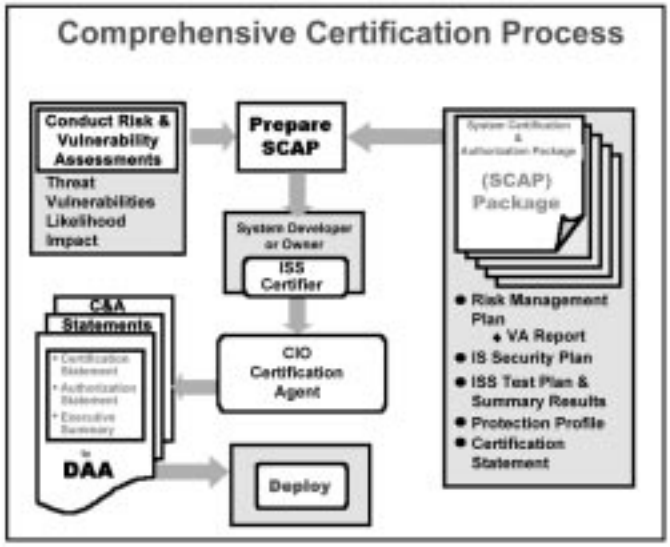


Figure 3: Certification Process

Continued on page 20

Continued from page 19

limited access procedures. Security guards that detect threats and take corrective measures to prevent intrusions monitor these personnel and physical barriers. In addition, the new FAA ISS Architecture addresses the cyber barriers that are required for each system/subsystem and their planned insertion as the National Airspace System evolves. Within a facility, personnel, physical, and cyber barriers may be made up of multiple layers of protection. The amount of personnel, physical, and cyber barrier protection at a facility is dependent upon the criticality of service(s) provided. Services directly supporting the National Airspace System require more protection than services supporting the administrative functions of the FAA, but all systems and services are addressed by the ISS Program.

The Operational Model

It is not enough, of course, to determine that facilities "in isolation" are certified. After all, FAA systems are configured and operated in a decentralized environment that spans multiple geographic locations and physical facilities. Each facility provides one or more services that support the operation of the FAA in ensuring the safe and efficient movement of air traffic. Because each facility can operate independently, and interdependently, this decentralized operational approach protects the National Airspace System against a widespread failure.

In recognition of this increasing interconnectivity, the FAA has developed an Operational Model that includes layers of security in the form of "boundaries of protection." This Operational Model, as depicted in Figure 4, provides multiple layers or "boundaries" of protection from an overall National Airspace System perspective, as well as protection for the individual services provided at each facility.

Since the operation of the National Airspace System (see Figure 4) is highly dependent on information systems, each facility must create a cyber barrier to address the ever-increasing potential of cyber threats. Because each facility works interdependently, they need to share information electronically through local and wide area networks. They also need to exchange information with other organizations, including other government agencies, the airlines, and the public. Because these communications networks are pipes that cross the personnel and physical barriers, they force the establishment of an additional protection barrier. The cyber barrier will prevent unauthorized or unauthenticated electronic access to the critical infrastructure of the National Airspace System. These cyber barriers include firewalls, encryption techniques, authentication protocols, and Public Key Infrastructure technologies. The strength and depth of the cyber barrier used by the network and by the individual facility will be dependent on the nature of the service and systems that are being protected.

Assuring the integrity and effectiveness of a cyber barrier to a facility's information infrastructure requires constant monitoring and analysis, both at the facility level as well as the national level. If an incident is detected, action is required at both the facility level and possibly at other facilities as well. Cyber threats have the potential to replicate themselves within a facility, as well as over an entire network. To respond to this threat, the FAA has developed both a local monitoring process and a centralized Computer Security Incident Response Capability (CSIRC).

Figure 5 provides the holistic perspective, presenting all three models—structural, process, and operational—and showing how they are interlinked to provide reinforcing elements of protection for the National Airspace System.

Cyber-Security Pyramid

The Structural Model, the ISS pyramid in Figure 1, has helped us to focus our ISS efforts. This model is based on five clearly defined layers of protection that have day-to-day meaning to the 48,000 employees of the FAA. In order to maximally benefit from the vast array of work being described in the computer security literature; however, there was a need to map our five layers of security into such standard cyber goals as confidentiality, authentication, integrity, access control and availability. By mapping these goals to the layers of our pyramid to which they were most closely aligned, we found a very natural, almost

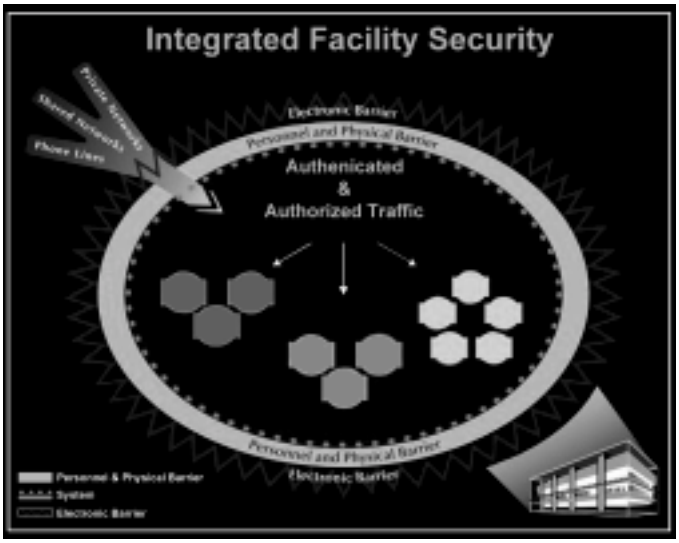


Figure 4: Integrated Facility Security

"isomorphic", relationship between the original structural model and the layered cyber model (Figure 6).

The top level of the pyramid, personnel security, easily maps to the protection mechanism of authentication. In personnel security, we want to assure that each and every user has a background check and/or security clearance and is known as a trustworthy member of the team who is allowed to use the systems. In a cyber-security sense this is equivalent to saying that we have ***Authenticated*** or verified not only all the users, but also the processes and tasks executing in the system.

The next level of the original pyramid was physical security. At that level we want to ensure that the FAA facilities are safe from unauthorized access and harm. This relates directly to the cyber-security protection mechanisms called ***Access Control***.

The third level of the ISS Protection Model is called Compartmentalization and Information Systems Security. This level provides the mechanisms to constrain and control the impact of any single security incident. For example, FAA's ATC facilities are resilient when a security incident at one facility cannot spread to another. In cyber-security terminology this is equated to ***Integrity***. By Integrity, we mean that data and systems cannot be altered without detection.

The fourth level of the ISS Protection Model is called Site-Specific Adaptation. At this level, we find that FAA systems have been designed in a general format, however, they have no operational capability until they have been adapted with a "unique" fingerprint of the airspace, geography, equipment, and procedures that makes each facility's ATC system work properly. In some sense, this is the "classified" data of the system that makes it work. So this relates closely to the cyber-security functions of ***Confidentiality***. By confidentiality we ensure that sensitive or classified information is neither available nor disclosed to unauthorized parties.

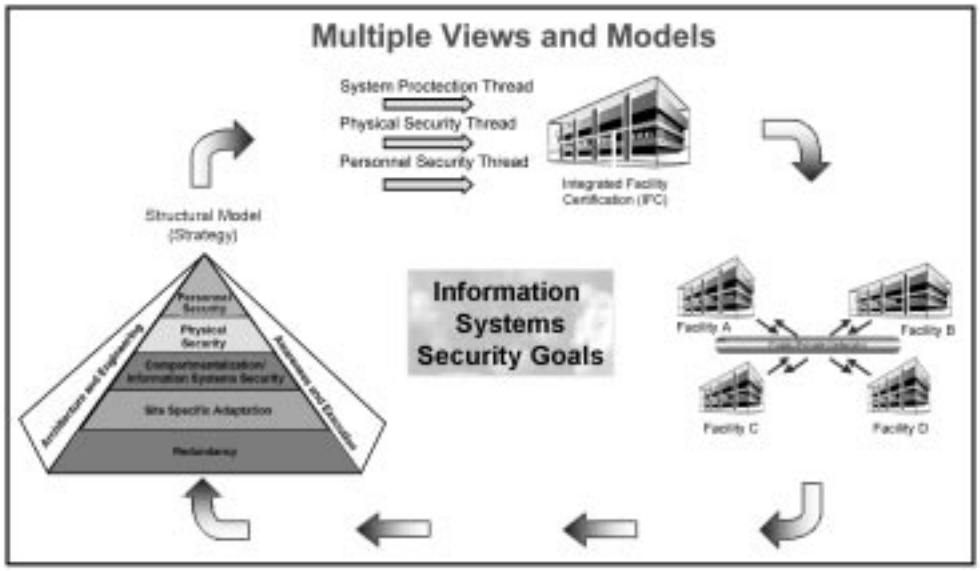


Figure 5: Interlinked Protection

The final level of the pyramid is called Redundancy. At this level, the system provides a degree of robustness that ensures that FAA systems perform as expected even if parts of the systems or sensors become unavailable or become corrupted by some means. Again, this relates directly to the cyber-security functional level of ***Availability*** that ensures that a resource is accessible and usable on demand by any authorized party.

The Cyber Model and the original Structural Model (shown side by side in Figure 6) provide the FAA with important and easy to understand perspectives for its ISS Program, and we believe they will be useful in explaining our objectives to a variety of audiences. The structural model has meaning to the lay people of the FAA; the cyber security view permits us to benefit from the efforts of the ISS experts at the cutting edge of this field.

Information Systems Prioritization

Although public awareness of the need for security in computing systems is growing rapidly, current efforts to provide security are unlikely to succeed if the goals and objectives for security are not well defined and prioritized. Many current security efforts suffer from the flawed assumption that adequate security can be provided in applications based upon one concept of security, based on the principal of "Need to Know." When we started to address our needs and priorities at the FAA, we noticed that the primary goals of "Need to Know" were for confidentiality and access control, rather than the FAA's needs to assure the integrity of our data and processes and at the same time assure availability of service. These needs are important to every system designer, but the current models for building trustworthy systems are derived from an earlier mathematical model called the Bell-LaPadula model, which addressed primarily the confidentiality of information. This model is described in many books on computer security and formed the foundation for the

Continued on page 22

Continued from page 21

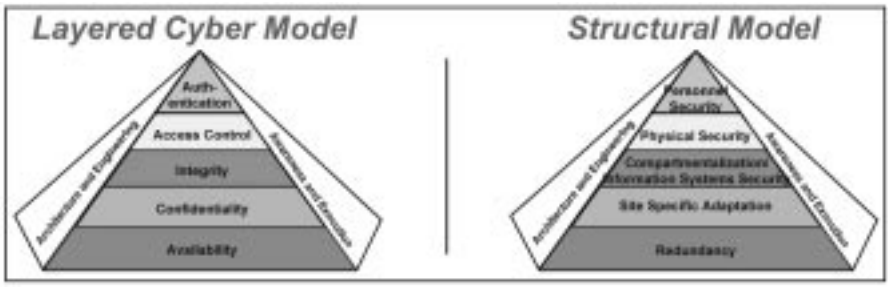


Figure 6: Layered vs. Structured Models

original "Orange Book" on Trusted Computing Systems [3].

Of growing concern with regard to controlling critical infrastructures are denial-of-service attacks, which compromise availability, and attacks on the integrity of the systems and their information. Experience has taught that systems—in particular complex systems—can be secure, but there will always be residual vulnerabilities. The question one should ask is not simply whether a system is secure, but how secure that system is relative to some perceived threat. Thus, notions of absolute security, based on correspondence to formal models, have to be enhanced to deal with the current environment where security relative to perceived threat is paramount.

Conclusion

The FAA has developed a holistic and multidimensional approach to ISS. The perspectives of our approach embrace multiple viewpoints including Structural, Process and Operational models. The models are all interrelated and are critical to the overall success of an effective ISS Program. As is the case in any rapidly developing field, these models and our approaches to them will change with time and with the threats. Additionally, we have noted that existing procedures based on the commonly used "Need to Know" model do not fully address the highest needs and priorities of the FAA. We are currently addressing these in several studies and will provide a more detailed mathematical and engineering analysis of these findings in the future.

Biography

Dr. Mehan joined the Federal Aviation Administration (FAA) in February 1999 as its first Chief Information Officer (CIO). He is the agency's principal advisor on information technology and directs strategic planning for information technology across the agency. Dr. Mehan also oversees the implementation of the FAA's Information Systems Security Program.

Before joining the FAA, Dr. Mehan held positions at AT&T for 31 years. At the time of his retirement, he was International Vice President of Quality and Process Management, leading efforts to align and optimize the processes and systems of AT&T's global operations. Prior positions held within AT&T included International CIO and Commercial Market Services Vice President, Product and Market Management Vice President for General Business Systems, Material Management and Customer Service Director. He began his AT&T career as a member of the technical staff at Bell Telephone Laboratories.

Dr. Mehan graduated from Drexel University with a bachelor's degree in Electrical Engineering. He holds a master's in systems engineering and a Ph.D. in Operations Research from the University of Pennsylvania. He has served on the boards of the U.S. Telecommunications Training Institute, the North American Telecommunications Association, and the Japan-U.S. Telecommunications Research Institute.

Biographical Summary

1995-1998 - International Vice President-Quality and Process Management, AT&T

1992-1995 - International Chief Information Officer (CIO) and Commercial Market Services Vice President, AT&T

1987-1992 - Product and Market Management Vice President for General Business Systems, AT&T

1984-1987 - Material Management and Customer Service Director, AT&T

1979-1984 - Division Manager Strategic Planning and Product Technical Support, AT&T

References

1. Mehan, Daniel J., Information Systems Security: The Federal Aviation Administration's Layered Approach, TR News, No. 211, Nov-Dec 2000, p. 8

2. Schneider, Fred B (editor), Trust in Cyberspace, National Academy Press, 1999, pp. 13-14

3. Pfleeger, Charles P., Security in Computing, Second Edition, Prentice-Hall, 1997, p. 278

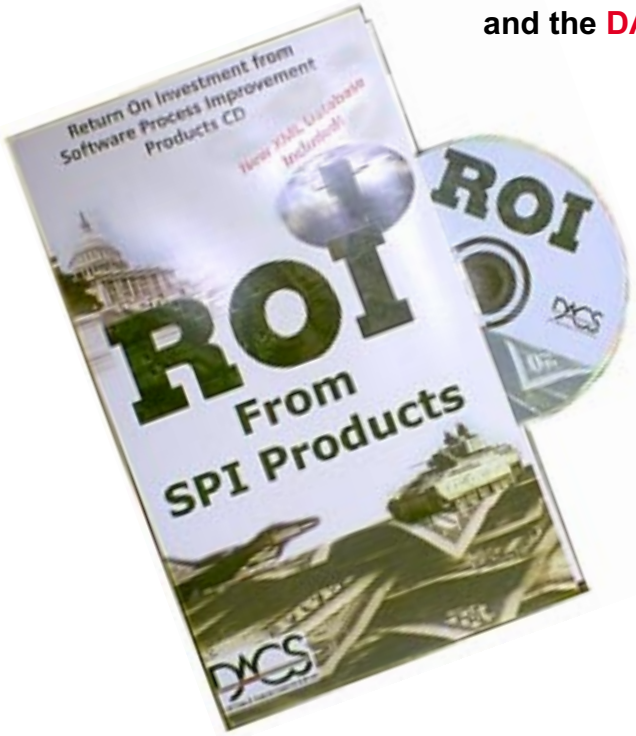
Contact Information

Dr. Daniel J. Mehan
Chief Information Officer
AIO-1
FAA Headquarters, Room 725
800 Independence Ave., SW
Washington, DC 20591
Daniel.Mehan@faa.gov

New DACS Products Available!

Return-On-Investment from Software Process Improvement (ROI from SPI) CD

and the DACS Documents CD



Return-on-Investment from SPI CD ($125)

The DACS ROI CD includes four products, two tools; the ROI Database nad the ROI calculating spreadsheet, and two technical reports; "Measuring ROI from Software Engineering and Management", and "Using Cost Benefit Analyses to Develop SPI Strategies".

"Measuring Return on Investment from Software Engineering and Management"

This newly revised report examines the details necessary to demonstrate from a business, profit and loss, and senior management perspective the benefits of using software process improvement (SPI) techniques. The report generalizes and models the cost benefits one can achieve from SPI. The report examines the business implications of the primary benefits (i.e., cost improvements, schedule improvements, quality improvements) as well as secondary benefits (i.e., improved employee morale, higher customer satisfaction). This report includes a spreadsheet and instructions for calculating Return-On-Investment.

Return-On-Investment (ROI) Database

The DACS ROI Database v1.0 captures the benefits gained by software organizations that have undertaken improvements in software management via software process improvement. Results from in excess of 50 organizations are included in this database. Improvements in cost, schedule, reliability, employee morale, and customer acceptance resulting from these improvements have all been captured in this database. A user interface is also provided to search the database to allow users to choose goals (e.g., reduce cost, improve quality), improvements (e.g. ,inspections, cleanroom), and what period of time they wish considered for their query. The system will then collect that data and present the query results in an easy to understand format.

DACS Document CD ($250)

The Data & Analysis Center for Software (DACS) has placed all of its documents in Portable Document Format (PDF) on a single CD. This product includes the DoD/IT Acronym List, every DoD Software Tech News, every DACS Product and Services Brochure and all DACS Technical Reports produced before June 2000. On this CD is over $600 worth of reports alone! CD Features:

- Full Text and Keyword Search Capabilities
- Document Listing by Title and Type
- Adobe Acrobat Reader with Search Capabilities

Either of these CDs may be purchased from the DACS Customer Liaison:
(800) 214-7921, cust-liasn@dacs.dtic.mil,

on-line at www.dacs.dtic.mil/forms/orderform.shtml,

or by mailing in the Order Form included in this newsletter.

Article Reproduction

Images and information presented in these articles may be reproduced as long as the following message is noted:

"This article was originally printed in the *DoD Software Tech News*, vol. 4, no. 3.
Requests for copies of the referenced newsletter may be submitted to the following address:

Data & Analysis Center for Software
Attn: Lon R. Dean, Editor
PO Box 1400
Rome, NY 13442-1400
(800) 214-7921; Fax (315) 334-4964
news-editor@dacs.dtic.mil

An archive of past newsletters is available at
www.dacs.dtic.mil/awareness/newsletters/."

In addition to the copyright message, we ask that you send us three copies of any document that references any article appearing in the *DoD Software Tech News*.

Thank you for your interest in the products and services of the Data & Analysis Center for Software (DACS).

Data & Analysis Center for Software
P.O. Box 1400
Rome, NY 13442-1400

PRSRT STD
U.S. Postage
P A I D
Permit #566
UTICA, NY

Return Service Requested