Defense Software Summit Report

(Final Draft, Version 1.0)

October 2006
Washington, DC

Office of the Deputy Under Secretary of Defense for
Acquisition and Technology
Systems and Software Engineering
Pentagon, Washington, DC 22310

in cooperation with the
Systems Engineering Division
National Defense Industrial Association

This report is maintained by the Office of the Deputy Under Secretary of Defense for Acquisition and Technology (ODUSD(A&T)), Systems and Software Engineering/Software Engineering and System Assurance (SSE/SSA). To submit questions, concerns, or recommendations about this report or to request a copy, please contact the office by e-mail, ATL-SSA@OSD.mil, or by telephone, 703-602-0851, extension 101. Please provide your contact information including telephone number for response. Copies of the report also will be available from the National Defense Industrial Association Web site: www.ndia.org.

CONTENTS

Preface 1
1 Introduction 2
2 Keynote Address 2
3 Program Executive Officer and Session Plenary Speaker Perspectives 3
3.1 PEO Panel: "Perspectives on Software-Related Acquisition Issues" 3
3.2 Session Plenary Speakers 3
3.3 Summary of Perspectives 4
4 Multi-Service and Defense Agency Panel 5
5 Software Issues Summary 6
5.1 National Defense Industrial Association Top Software Issues 6
5.2 Recommended Actions for Issue Resolution 6
6 Workshops 7
6.1 Software Acquisition and Sustainment Workshop 7
6.2 Software Policy Workshop 9
6.3 Human Capital Workshop 12
6.4 Software Engineering Practices Workshop 15
7 Preliminary Analysis 19
8 Closing Remarks and Next Steps 20
Concluding Material 21

This page intentionally left blank.

Preface

The Deputy Under Secretary of Defense for Acquisition and Technology hosted the Defense Software Summit in Washington, DC, on October 18–19, 2006. The purpose of the summit was to identify the current situation, issues, barriers, and recommendations concerning Department of Defense software development. The summit provided a forum for guest speakers to present information and for attendees to conduct workshops.

The summit raised awareness of significant software engineering and management issues within the Department. Summit participants helped identify specific actions the Department needs to take to reduce or eliminate adverse situations and to solve problems.

As a starting point for workshop efforts, the summit hosts presented the following seven issues, derived from an August 2006 report by the National Defense Industrial Association. Each workshop reviewed and expanded upon these issues.

1. The impact of system requirements upon software is not consistently quantified and managed.

2. Fundamental systems engineering decisions are made without full participation of software engineers.

3. Software life cycle planning and management by acquirers and suppliers are ineffective.

4. The quantity and quality of software engineering expertise are insufficient for dealing with complex modern systems.

5. Traditional software verification techniques are costly and ineffective for dealing with complex modern systems.

6. There is a failure to ensure correct, predictable, safe, secure execution of complex software in distributed environments.

7. Inadequate attention is given to total life cycle issues for [software] commercial off-the-shelf/non-developmental item impacts on [total system] life cycle cost and risk.

(Source: National Defense Industrial Association Top Software Issues, August 2006)

The workshops produced recommendations to correct these issues. Workshop participants also identified additional issues and corresponding recommendations for their areas. The summit provided an excellent start to addressing Department of Defense software issues. The observations and recommendations provide the Department with a solid foundation for corrective action and for future software life cycle management strategy.

1 Introduction

This report summarizes results of the Department of Defense (DoD) Software Summit, Washington, DC, October 18–19, 2006. The report is organized according to the meeting events. The report includes a brief description of the software engineering, acquisition, and management conditions within DoD.

Defense Software Summit Event Outline

- Keynote Address: The Honorable Dr. James I. Finley, Deputy Under Secretary of Defense for Acquisition and Technology, followed by Mr. Mark Schaeffer, Director, Systems and Software Engineering, ODUSD(A&T)
- Program Executive Officer Panel: "Perspectives on Software-Related Acquisition Issues"
- Multi-Service and Defense Agency Panel: "Service and Agency Software Initiatives and Strategies"
- Session Plenary Speaker Topics: National Defense Industrial Association (NDIA) Top Software Issues, Software Industrial Base Study, and Software Producibility
- Workshop Introductions: Ms. Kristen Baldwin, ODUSD(A&T), Software Engineering and System Assurance
  - Software Acquisition and Sustainment, led by Mr. Mike Nicol, Air Force Aeronautical Systems Center and Mr. Lawrence T. Osiecki, U.S. Army, Armament Software Engineering Center
  - Software Policy led by Mr. James Clausen, DoD Chief Information Office, Office of Commercial Information Technology Policy; Col. Peter Sefcik, Jr., USAF, Chief, Air Force Engineering Policy and Guidance Team; and Lt. Col. Mark Wilson, SAF/AQR Systems and Software Engineering
  - Human Capital, led by Dr. Kenneth E. Nidiffer, Fellow, Systems and Software Consortium and Mr. George Prosnik, Defense Acquisition University E&T Center
  - Software Engineering Practices, led by Mr. Grady Campbell, Software Engineering Institute and Mr. Paul R. Croll, CSC, Industry Co-Chair, NDIA Software Committee

2 Keynote Address

The Honorable Dr. James I. Finley, DUSD(A&T), hosted the Defense Software Summit. In his keynote address, he presented his view of the software challenge, stating that the Department of Defense (DoD) needs to do the following:

- Reshape the [acquisition] enterprise using short- and long-term initiatives that accelerate lasting change for elements of the acquisition system.
- Strive to reduce cycle time, improve communications, and increase competitiveness.

- Strive to achieve acquisition excellence through a Systems and Software Engineering Center of Excellence that will facilitate collaboration, integration, and information sharing.
- Focus on software: The demand for software is increasing, requirements are expanding, and programs face many software performance, schedule, and cost issues.
- Reduce or eliminate adverse software trends, including poor requirements, immature architectures, incomplete planning, poor scheduling, and vague metrics.

Dr. Finley presented four objectives for the Defense Software Summit participants:

1. Characterize the situation.
2. Identify key issues.
3. Describe barriers to implementing recommendations.
4. Recommend actions.

Mr. Mark Schaeffer welcomed participants and endorsed Dr. Finley's challenges and objectives for the summit. Mr. Schaeffer also invited participants to share their insights, experiences, best practices, and recommendations concerning software acquisition, engineering, and resources because this information will be used to support development of policy and guidance and will help focus resource application within the Department.

3 Program Executive Officer and Session Plenary Speaker Perspectives

3.1 PEO Panel: "Perspectives on Software-Related Acquisition Issues"

Following the keynote address, the Program Executive Officer (PEO) panel presented its views of software development challenges. The panel consisted of the following members: Lt. Gen. Charles L. Johnson II, Commander, Electronic Systems Center, Hanscom Air Force Base / PEO for Command Control and Combat Support; Lt. Gen. Michael A. Hamel, Commander, Space and Missile Systems Center, Air Force Space Command, Los Angeles Air Force Base / PEO for Space; Brig. Gen. Nick Justice, Deputy PEO, Command, Control, Communications–Tactical; and RADM Craig E. Steidle, U.S. Navy (Retired), Professor-Aerospace Engineering, U.S. Naval Academy / Former Director, Joint Strike Fighter Program.

3.2 Session Plenary Speakers

Keynote speakers presented information of interest. Mr. Pierre Chao, Senior Fellow, Center for Strategic and International Studies, presented the Software Industrial Base Study. Mr. Robert Gold, Associate Director for Software and Embedded Systems, Office of the Director, Defense Research and Engineering, presented information about Software Producibility. Mr. Geoff Draper, Harris Corp., presented the NDIA Top Software Issues. The issues are presented in Section 5, Software Issues Summary.

The PEO panel and plenary speakers summarized some of their agencies' software-intensive projects and shared their perspectives concerning DoD software issues.

3.3 Summary of Perspectives

- Problems
  - Improved systems and software engineering methods may reduce problem root causes and provide $24B in cost avoidance over the DoD Five-Year Defense Plan.
  - Problem root causes include lack of requirements discipline, limited staff experience, external adverse influences, poor planning, resource and budget constraints, weak contract management, funding turbulence, inadequate program documentation, and incomplete risk management.
  - Some executives have a difficult time with software management issues.
- Situation
  - The DoD software science and technology (S&T) condition is problematic: Research is declining; government expertise has atrophied; the Department has limited programs for tool development; there is no multi-Service approach to S&T issues; and there is a long-standing assumption, without confirmation, that industry will solve or address all software issues.
  - Decision making and analysis are hampered by a lack of quantitative data about large-scale software programs.
  - Software and system development tool communities lack a consistent vision.
- Tools
  - Software development tools do not adequately provide system development awareness of progress, design completeness, requirements traceability, and testing.
  - Tools need to complement people, support reuse, provide status, and simplify testing.
  - Programs would benefit from improved tools to manage data and knowledge, to foster interoperability, to address system of system design, and to assist in verification.
  - Programs need an approach to develop or identify technologies, tools, and standards for large-scale systems.
- Management Ideas
  - Programs are shifting from building software systems to composing systems using commercial and otherwise available software.
  - There is a trend to focus on a life cycle phase, such as development. Need to synchronize the total life cycle support.
  - Need to ensure data visibility: label it, find it, sort it, store it, protect it, maintain it, and share it while maintaining security awareness.
  - Size of requirements drives speed of delivery; produce smaller, more manageable modules.
  - Test incrementally, thoroughly, and quickly to reduce program cycle time.

- Security
  - Need to recognize that adversaries have access to information technology (IT).

4 Multi-Service and Defense Agency Panel

Representatives from the Army, Air Force, Navy, and National Security Agency shared their insights regarding defense initiatives and plans.

Panel members were: Mr. Carl Siel, Chief Engineer of the Navy, Office of the Assistant Secretary of the Navy, Research, Development and Acquisition; Mr. Terry Jaggers, Deputy Assistant Secretary for Science, Technology & Engineering, Air Force; Mr. Kelly A. Miller, National Security Agency/Central Security Service, Principal Director of Engineering; and Mr. Doug Wiltsie, Office of the Assistant Secretary of the Army for Acquisition, Technology and Logistics. The Department of Defense needs to do the following:

- Establish strategic initiatives for acquisition process improvement, program measurement, training and education, architecture, system of systems integration, analysis, and planning.
- Perform systems and software engineering over the life cycle.
- Integrate software acquisition in make or buy frameworks.
- Understand that software is a primary performance, schedule, and cost driver; recognize that software engineering is inseparable from disciplined systems engineering; and define a capability engineering framework.
- Remove cultural barriers between information technology and weapon software developers; and need to track and manage the health of system and software engineers.
- Increase leadership awareness; improve engineering practice and discipline; and develop and retain a skilled work force.
- Exploit reuse to reduce cost and schedule; promote architecture; and use open architecture and product line approaches.
- Establish centralized policy; integrate systems and software engineering; and provide guidebooks for Product/Project/Program Managers (PMs).
- Address acquisition management, engineering, development techniques, business implications, and human resources.
- Establish near-term tasks including the following: Conduct “proof of concept” Lean Six Sigma software-intensive projects; incorporate process improvements where feasible; provide request for proposal preparation for software-intensive systems guidance; collect metrics; improve quality; develop and conduct a software leadership course; and establish and use software product lines.
- Improve software estimating.

5 Software Issues Summary

5.1 National Defense Industrial Association Top Software Issues

Mr. Geoff Draper, Harris Corp., presented the NDIA Systems Engineering Top Software Issues report. The NDIA industry and defense software issues and recommendations follow.

National Defense Industrial Association Issues

- The impact of system requirements upon software is not consistently quantified and managed.
- Fundamental system engineering decisions are made without full participation of software engineers.
- Software life cycle planning and management by acquirers and suppliers are ineffective.
- The quantity and quality of software engineering expertise are insufficient for dealing with the complexity of modern systems.
- Traditional software verification techniques are costly and ineffective for dealing with the complexity of modern systems.
- There is a failure to ensure correct, predictable, safe, secure execution of complex software in distributed environments.
- Inadequate attention is given to total life cycle issues for commercial [software] off-the-shelf/non-developmental item (COTS/NDI) impacts on [total system] life cycle cost and risk.

5.2 Recommended Actions for Issue Resolution

NDIA recommendations followed the issue summary.

National Defense Industrial Association Recommendations

1. Enforce effective software requirements development and management practices, including assessment of change impacts for both the acquirer and the supplier organizations.
2. Institutionalize the integration and participation of software engineering in all system engineering activities.
3. Establish a culture of quantitative planning and management, using proven processes with collaborative decision making across the software life cycle.
4. Collaborate on innovative strategies to staff to appropriate levels and to attract, develop, and retain qualified talent to meet current and future software engineering needs in government and industry.
5. Study current software verification practices in industry, and develop guidance and training to improve effectiveness in ensuring product quality across the life cycle.
6. Collaborate with industry to develop approaches, standards, and tools, addressing system assurance issues throughout the acquisition life cycle and supply chain.
7. Improve and expand guidelines for addressing total life cycle COTS/NDI issues.

6 Workshops

Attendees further examined the PEO and Service main points, NDIA issues, recommendations, and session plenary speaker perspectives through the following four workshops:

1. Software Acquisition and Sustainment
2. Software Policy
3. Human Capital
4. Software Engineering Practices

Each workshop panel was asked to describe and identify the following for its topic:

- Situation
- Additional issues
- Barriers
- Recommendations

Following are summaries of workshop results.

6.1 Software Acquisition and Sustainment Workshop

The software acquisition and sustainment workshop was designed to explore:

- Acquiring software in a systems acquisition context
- Life cycle management of software requirements
- Expectation management of the life cycle events
- Progress checking and tracking
- Size, complexity, and resource estimating

These topics combined with the other summit information helped form the basis of the following software acquisition and sustainment situation, issues, barriers, and recommendations.

Software Acquisition and Sustainment Situation

- Need flexible acquisition approaches to accommodate a large range of software challenges.
- Large, revolutionary systems entail more risk and unpredictable program challenges.
- Requirements are not fully defined at program start; they seem to evolve.
- Expectations about software-driven capabilities are established without adequate knowledge of requirements, technology, or development complexities.
- Management has limited visibility into the software development process and status.

- Acquirers do not adequately address sustainment and total life cycle during early program phases.
- Risk areas: Single point failures are not adequately addressed, such as single providers of software, key personnel stability, insufficient data rights, or life cycle support of COTS.
- There are decreasing numbers of trained, experienced software acquisition personnel, which limits the Department's ability to establish executable programs and monitor execution.

Software Acquisition and Sustainment Issues

- Programs knowingly and unknowingly overcommit program abilities and capabilities.
- Software issues are not addressed early enough in the life cycle.
- Programs lack robust or detailed system and software architectures.
- Acquisition programs are started with system design and demonstration contracts before necessary systems and software engineering are accomplished.
- System engineering milestones or events may impede incremental software developments.
- Programs lack uniform application of metrics and earned value management to software.
- Program offices are not addressing the life cycle implications of COTS and GOTS (government off-the-shelf) software.

Software Acquisition and Sustainment Barriers

- DoD lacks adequate numbers of trained and experienced government systems and software engineers.
- Conflict: Demand for software is increasing, while the number of software acquisition personnel continues to decline; the course of corrective action is unclear.
- Declining software development capability challenges awareness and funding stability.
- There is a one-size-fits-all acquisition process within the acquisition community.
- Program Objective Memorandum and budget processes require program budget definition before the definition of crucial details.
- Warfighters are suspicious of incremental approaches because the second or third increment may not arrive.
- Procurement system requires too many approvals.
- Difficult to decompose a large system into smaller ones (the engineering process takes time).
- Sustaining multiple baselines is challenging and diverts resources.
- Historical trends indicate that award or incentive fees are based on schedule rather than on performance. This approach detracts from improving product quality.

Software Acquisition and Sustainment Recommendations

For software-intensive, mission-critical, or high level of mission software in systems:

- Establish categories of systems based on the software challenge and manage the development and expectations accordingly. Redefine the acquisition process, as needed, so that different approaches and expectations are established based on the type of development, such as the following: highly unprecedented, extensions of current systems, or commercially available.
- Identify a core set of software metrics and improve reporting.
- Establish evolvable systems and software architectures.
- Require Earned Value Management reporting at all levels, including the software level.
- Improve life cycle coverage of software in the Systems Engineering Plan.
- Emphasize acquisition "ilities" such as functionality, producibility, supportability, executability, affordability, and reliability, for software.
- Establish an infrastructure that provides "no-cost" support, such as providing advice, conducting program reviews, and sharing lessons learned, to the project management office.

To overcome the lack of attention by acquirers regarding sustainment issues:

- Require software sustainment planning at all milestones: By Milestone B, plan for sources of support, and transition from development to support, fielding, funding, training, documentation, and technology refresh/upgrades.
- Address software legal issues, such as intellectual property and licenses.
- Understand the architectures and associated life cycle effects.
- Address sustainment in the Systems Engineering Plan.

To improve the government's ability to perform smart buying:

- Develop and maintain a software workforce competency maturity plan, a.k.a., "roadmap," that describes key factors, such as decisions, competencies, management, research, and development.
- Balance software acquisition workforce levels with required capabilities or define plan to achieve required capabilities with personnel shortages.
- Include the life cycle effects of COTS software.
- Improve risk management over the software life cycle.
- Improve the architecting of systems and software.

6.2 Software Policy Workshop

The software policy workshop started with the NDIA issues and recommendations and PEO, Service, and plenary speaker information. This workshop also considered the following:

- Current policies, related guidance, and policy effectiveness
- Policy consistency in DoD
- Implementation of Defense Science Board and other study recommendations
- Defense and industry software standards

The workshop produced situation, issues, barriers, and recommendation information. A summary follows.

Software Policy Situation

- Current policies have been established with the best of intentions, but many PMs and developers see the negatives about the situation rather than the positive aspects.
- Currently some services, such as the Navy, are requiring software development plans in their programs; this is a step in the right direction.
- Selected agencies have repositories of policies and practices to help share information, but consideration of "COTS software first" to satisfy mission needs is inconsistent across the Department.
- A common portal is needed to support current software policy; the portal would need to include policies and guidebooks searchable by any term, samples, lessons learned, artifacts, frequently asked questions, and possibly an "ask an expert" feature.
- The Department needs a software resource analysis, policy expert, and integration team with sufficient expertise to oversee and implement policy.
- Policy implementation mechanisms (instructions, manuals, etc.) are missing; some policies are approved without guidelines, standards, or training.
- Weapon system and Chief Information Officer policy issues are sometimes too diverse and too large for any single component to address adequately. There is a significant potential for second- and third-order consequences if handled at component level because of DoD-level ownership and impacts of existing policy.

Software Policy Issues

- Current policies are onerous because of the cost of compliance. Resource needs are high.
- Some policies are redundant. Policies are often difficult to locate.
- There is a lack of standards and guidelines to implement policy. Limited or no training about policies hampers implementation and compliance.
- There are problems with enforcement. Examples: Programs will not follow policy if the leadership does not show an interest. Submission and quality of software data and cost reports vary; these reports are required but are difficult to evaluate, and it is difficult to enforce accurate reporting.
- There is a gap in policy evident through the examination of the Department of Defense Acquisition Guidebook chapters 4 and 7.

- There may be a conflict between earned value management objectives and fixed price contracts.
- Limited expertise within the Office of the Secretary of Defense (OSD) hinders software technical review and oversight.
- OSD leadership does not own or manage a common portal to share information. This limits fast and accurate information sharing.

Software Policy Barriers

- Materiel developer's lack of resources: time, money, people, and command priority.
- The Department lacks a primary office for software life cycle matters.
- Many within the acquisition community are ignorant of policy, guidance, and best practices, especially when those practices come from the commercial software or systems integration communities.
- Job security: People tend to share only their successes and avoid publicizing negative information.
- Information security has affected or closed down some of the portals, which impedes access to information.
- Conflicts exist surrounding information sharing. Some developers may not be willing to share best practices because of concerns about protecting proprietary information and keeping a competitive advantage.

Software Policy Recommendations

To facilitate information sharing, create within 2 years a common portal that will:

- Identify the owner and location of the portal.
- Contain current and accurate material, such as policies, standards, guidebooks, instructions, tools, memoranda, and lessons learned.
- Provide special features, such as frequently asked questions, "ask the expert," and a robust search engine to find information.

To refine and streamline policy and directives within an estimated 2 years:

- Initiate a study, at OSD level with Service support, to examine links between information technology policy and weapon system acquisition policy, and recommend any necessary actions (1–2 years). This effort will help reduce dual oversight, analysis, and policies in some areas.
- Plan to analyze DoD and Service software policy.
- Cross-reference analysis results, including information technology and weapon systems.
- Compare policy between information technology and weapon systems to determine similarities and differences.

- Define common policy elements across Services and defense agencies.
- Draft updated policy based on common themes.
- Quantitatively evaluate policy before large-scale implementation, acceptance, or approval.
- Align policy with procedures and processes where applicable.
- Address policy span of control because some policies are too broad to conveniently manage.
- Address Department policy second- and third-order effects on the Services and PMs to determine and prevent unintended consequences.

To assist Program Executive Officers, Project Managers, Product Managers, and others in policy and best engineering and acquisition practice compliance within 1 to 2 years:

- Create and empower an organization to assist PEOs and PMs regarding compliance and professional best practices.
- Create an independent software group of experts to collect, analyze, synthesize, and share engineering and management information about reliability, issues, schedules, cost, history, forecasting, metrics, and other appropriate areas.
- Require program offices to have software engineering expertise.
- Update DoD guidance to ensure that work breakdown structures improve software viability in acquisition processes such as earned value, cost, and schedule reporting.
- Update guidance in the areas of COTS, software best practices, portfolio management, and risk management as a minimum.

To assist the acquisition community, the Department needs to balance:

- Data sharing and security needs.
- Commercial practice versus specifications and standards.
- Statements of work/objectives and the additional level of specificity to development software.
- Development, modified, COTS, GOTS, and reuse software guidelines.

6.3 Human Capital Workshop

The Human Capital workshop addressed two key topics and associated question sets.

- Recruitment and retention: What are near-term and long-term workforce capability risks? What capabilities are needed to execute the mission?
- Education, training, and competency development: Which competencies are mission critical? Where do individual and organizational competency gaps exist?

Human Capital Situation

- There are challenges in recruitment and retention, acquisition workforce downsizing, education and training, competency development, and the implications of offshore software development.
- Demand for personnel with security clearances is increasing, while the level of granting of clearances seems to remain static.
- At many organizational levels, the Department lacks highly experienced managers, system and software architects, domain experts, and technical experts (so-called "golden collar" personnel).
- Depth and breadth of personnel experience within the DoD acquisition base are eroding.
- Competition among industry and government affects the availability of key personnel.
- Information and knowledge sharing approaches and value sets differ among generations within the workforce.
- Based on college enrollment and choice of majors, within the United States, many young people seem to consider systems engineering and software engineering as career dead ends because of a perceived lack of potential for professional growth in these fields.
- An emerging technical and management skill set, for which some skills are still being discovered and defined, may be required for future complex DoD systems, such as systems of systems.

Human Capital Issues

- Because the personnel security clearance process is lengthy, delays in attaining security clearances affect software development efforts and capacity, especially for smaller companies without a large capital or project base.
- Because of DoD hiring freezes, the younger and older bimodal workforce distribution has resulted in a lack of personnel midgrade mentoring. This lack of mentoring adversely affects knowledge transfer and transition.
- Experienced systems and software engineers are missing from key leadership positions within the acquisition corps.
- "Generation Y" and younger personnel sometimes exhibit communication and work styles that differ from those of the majority of current managers in agencies and chains of command[1]. These differences can affect the workplace and office culture and can have a negative impact on operations.

[1] Cultural changes are needed in some authors' views to effectively manage younger generation software developers. See for example, "IT Managers Told to Think Young," Patrick Thibodeau, *Computerworld*, October 16, 2006, p. 14.

Human Capital Barriers

- Small companies may not be able to bear the costs of obtaining personnel clearances. This difficulty may hinder company participation in some contracts or work efforts[2].
- Generational knowledge transfer is impaired because lessons learned embedded within process directives are lost during acquisition reform. Reforms tend to restrict process and standard development and use.
- There are perceived hiring issues in filling Department acquisition midlevel slots. Certification, clearance, or military domain experience may be perceived as a hiring barrier because these qualifications may be difficult to find among non-DoD personnel.
- Outsourcing as a mitigation approach is not viable because DoD security policies delay hiring.

Human Capital Recommendations

- For workforce mid-career experience gaps, build experience bases and orient them to senior personnel. Assist mid-careerists and knowledge transfer by encouraging use of such knowledge transference approaches such as:
  - Communities of practice
  - Performance-based tools
  - Preferred/prescriptive process standards in key areas

For DoD acquisition corps careerists:

- Establish a software engineering specialty track within the acquisition field.
- Emphasize software acquisition management skill for Systems Engineers and Project Managers.
- Address apparent lack and/or low numbers of technical software personnel currently identified to fill key leadership positions within the DoD Acquisition Corps.
- Assess role, certification levels, and skill base of contractors and government program offices.

To improve personnel effectiveness:

- Assess the results of ongoing service and agency human capital and Section 804, Software Management Improvement, initiatives; and select best efforts for implementation.[3]

[2] For instance, the DoD Security Clearance Program and impacts of processing delays was highlighted again by the GAO in its most recent High Risk Listing (GAO-07-310). This area was first designated as a high-risk area in 2005. In January and February 2006, the clearance cycle averaged 419 days for 2,259 industry personnel surveyed vs. DoD goal of 180 days.

[3] Section 804 of the FY03 Defense Authorization Act (now Public Law 07-314) emphasizes software acquisition process improvement. The focus is on acquisition planning, requirements development and management, project management and oversight, and risk management. The Act requires metrics for performance measurement and process improvement, a process to ensure acquisition personnel have appropriate experience or training, and a process to ensure adherence to acquisition processes and requirements. Services and agencies have produced and implemented various action plans in response to this specific legislation.

- Encourage Project Management and Executive Office PEOs to embark on internal process improvements, using models such as CMMI-ACQ (Capability Maturity Model Integration–Acquisition).
- Simplify processes for obtaining clearances and assign priority to critical positions.
- Use acquisition management teams with individuals having expertise in both software development and systems engineering.

6.4 Software Engineering Practices Workshop

The Software Engineering workshop participants examined the NDIA issues and recommendations, and summit presenter information. They also examined:

- Synchronization of system and software engineering practices
- Assessment methods for software quality
- Commercial off-the-shelf software and open source issues, and
- General methods, tools, and techniques

The workshop efforts resulted in a summary of the situation, issues, barriers, and recommendations.

Software Engineering Practices Situation

Requirements generation

- Developers, customers, and acquirers do not address the impacts of changes to requirements consistently.
- Software requirements are not always adequately addressed in system-level requirements or solicitations and contracts.
- Weak linkage to software requirements for capabilities and portfolios may be an issue or barrier.
- Software requirements are only partially defined at the start of a program. The true understanding of software requirements is achieved as the system design evolves.

Life cycle issues

- Systems engineering and software life cycles, processes, and products are not always consistent or harmonized for meaningful cross-discipline integration to occur.
- System acquisition approaches and procedures do not leverage software's ability to rapidly implement improved capabilities.
- Software processes and methods are not always sufficiently followed when programs face budget or schedule pressures.

- Pressure to rapidly procure new capabilities can inhibit the balance of life cycle cost, schedule, and performance expectations to achieve executable programs.
- Developers provide inadequate attention to sustainment issues early in the life cycle.
- Software is not consistently involved in architectural decisions early in the life cycle. Software must evolve with the computing system architecture.
- Software risks and life cycle costs are not consistently accommodated in planning.
- Realistic schedule and effort or cost estimates are often rejected or constrained.
- Segregation of systems engineering and software proposal information may impede coordination among acquisition personnel. Software is treated as a separate entity.

Security and assurance

- Assurance of systems and assurance of a system of systems cannot be easily inferred from components because of issues such as emergent behavior or concurrent effects.
- Software is inherently vulnerable to widespread information [security] assurance threats; there is a need to ensure confidence in the supply chain pedigree.

Personnel and culture

- Acquisition program career path incentives are insufficient to attract software professionals.
- Education, training, and certifications are inadequate to ensure effective test skills.
- Staffing: There is a limited number of software professionals with DoD subject area expertise to meet the growing demand because of inadequate funding, insufficient career incentives, competition, downsizing, etc.
- Many times delegation of authority is missing; people need authority to participate in system-level decision making and trades.

Measurement and evaluation

- Software measures are not used effectively or acted upon.
- There is an over-reliance on testing alone rather than on robust software verification techniques. There is a need for rigorous peer reviews and robust software verification techniques.
- Compliance-based tests do not adequately cover risks or failure conditions.
- Exhaustive testing to rule out vulnerabilities is not feasible.
- Current techniques are inadequate to verify assured components with well-understood properties.

Commercial off-the-shelf and non-developmental items

- Reuse, open source, and government off-the-shelf software estimating methods are inadequate.

- COTS, NDI, and product-line best practices are known but not consistently implemented.
- Customer expectations for customization reduce benefits of COTS solutions.
- Open source licensing issues can expose organizations to liability, loss of data rights, and potentially extensive rework.

Software Engineering Practices Issues

- The impact of system requirements upon software is not consistently quantified and managed in development or sustainment.
- Conditions leading to software requirements changes are not clearly understood.
- Fundamental system engineering decisions are made without full consideration of software engineering.
- Software life cycle planning and management are ineffective; acquirers and suppliers need better software life cycle planning and management methods.
- The quantity and quality of software engineering expertise are insufficient to meet the demands of the government and the defense industry.
- Traditional software verification techniques are costly and ineffective for dealing with the scale and complexity of modern systems.
- There is a failure to ensure correct, predictable, safe, and secure execution of complex software in distributed environments.
- Inadequate attention is given to total life cycle issues for COTS/NDI impacts on life cycle cost and risk.
- More software expertise needs to be applied to planning and management.
- Software personnel staffing is inadequate across the system life cycle.
- Use of standard terminology needs improvement. There is too much jargon. Need a clear definition of “system of systems.”

Software Engineering Practices Barriers

- DoD, contractor, and project management office cultures or mind-sets are hard to change, causing clashes between groups and functional discipline areas.
- Resource limitation: Many good practices are sidestepped because of limited dollar resources.
- Developers lack the availability of validation of tools, such as Underwriters Laboratory.
- Developers also lack incentives to take intelligent steps to solve government and contractors’ software problems.
- The acquisition community does not seem to fully understand the incentive structure of the acquisition system.
- There are constraints on funding, such as congressional funding restrictions.

- Misalignment between program and contractor goals will cause severe problems.
- Leadership is still focused on hardware; many Defense Acquisition University courses are devoid of software; and there is a perspective that software is magic.
- The acquisition community has a diminishing experienced software workforce.
- Personnel procedures prevent a rapid transition of skilled staff to new programs.
- Acquisition methods have insufficient infrastructure (e.g., guidelines, procedures, and incentives) to create and facilitate reuse across organizations.
- Many leaders do not understand software or appreciate its challenge.

Software Engineering Practices Recommendations

To improve software engineering, the Department needs to implement or facilitate the following actions:

- Translate best practices into policy where needed.
- Revise old standards and update as needed.
- Review contracting guidelines to ensure currency and accuracy.
- Improve system engineering planning, introducing guidance resembling that of a computer resource life cycle management plan.
- Ensure continuity of engineering personnel for large-scale projects to reduce staff turbulence.
- Conduct independent program assessments to ensure that the program is following best practices.
- Encourage industry to request independent assessments as well.
- Team software and system engineers within a program.

Provide a cadre of experts to:

- Assist project start-ups to help balance performance schedule and cost among hardware and software performance, schedule, cost, supportability, security, and risk.
- Provide architecture expertise to support programs.
- Provide technical advice to help balance perspectives such as system, system of systems, family of systems, net-centricity, network, data sharing, interoperability, security, and transparency, among others.

Resource training and education to:

- Realistically create experts to satisfy personnel needs.
- Provide leadership training to enable informed decision making regarding software.
- Reinforce software engineering and management in the systems, planning, development, research, and engineering acquisition specialty.

Revise contracting methods and procedures to:

- Allow additional detail for software requirements if needed.
- Raise software as a major factor and improve visibility within the work breakdown structure for mission-critical or software-intensive systems.
- Clarify contractor role in conducting system engineering technical analysis for software integration.

Create or identify software tools to help manage the software development process:

- Create a DoD architecture framework that better represents software needs.
- Promote the use of product-line practices by acquisition programs and industry for the creation and use of cross-program capabilities and assets. Examine associated lessons learned regarding reuse.
- Identify tools to support decomposing large programs or architectures into smaller ones.
- Provide a DoD test bed or laboratory that provides persistent network test environments.
- At design reviews, encourage demonstrations; have developers show a capability or requirements satisfaction.

7 Preliminary Analysis

Workshop leaders analyzed the relationships among these workshop issues and found that:

- Program schedule pressure is perhaps the most common reason for failures and delays in the development of software-intensive systems.
- The following issues also contributed to failures and delays:
  - Lack of policy enforcement
  - Lack of systems and software process harmonization
  - Software process maturity/capability
  - Software review expertise that is just emerging within the Office of the Deputy Under Secretary of Defense for Acquisition and Technology.
  - Software issues and architecture not considered early enough in the system life cycle.
  - Lack of standards and systems engineering expertise in key leadership positions.

The summit's issues and recommendations provide an excellent start. Attendees achieved consensus on the seven initial NDIA issues, which, combined with the workshop issues and recommendations, provide DoD with a framework to create corrective action plans. Summit participants also acknowledged that 2 days are not enough time to produce a comprehensive list of issues and recommendations. Additional work is required to address issue resolution responsibilities, schedules, and resource needs. A follow-on summit with more time might result in a better prioritization of problem areas and development of more mature recommendations.

8 Closing Remarks and Next Steps

These topics warrant further review, analysis, and action. The Service representatives and workshops reinforced NDIA issue and action summaries as well as PEO and guest speaker viewpoints. The Department needs to continue to explore, assess the situation, determine if additional actions are needed to eliminate adverse situations, and take action to eliminate issues and reduce barriers. Suggestions for follow-on action:

Recommendations

- Synthesize summit results in detail. Examine and integrate workshop and overall summit findings.
- In detail, review and analyze Service, defense agency, and other selected agency, such as National Security Agency (NSA), software engineering, acquisition, and life cycle management initiatives, policies, processes, and plans.
- Publish the results of the detailed Service, defense agency, and summit analysis.
- Solicit Service, major command, engineering center, and PEO software life cycle management recommendations.
- Define and publish the Department's long-term objectives and course of action with associated priorities and resources in a software life cycle strategy.
- Track and monitor follow-on actions.
- Conduct a 2007 summit to review these actions and recommend adjustments.

The summit confirmed that software demand is increasing; managing software projects is difficult; availability of trained and experienced personnel is limited; standards and processes vary; and budgets are declining. The summit also confirmed that the Department software management staff has the will and desire to improve the software environment. Department action is needed. The Department needs to continue to aggressively focus on software engineering, acquisition, management, and human resource life cycle challenges through the application of resources and focused action.

Concluding Material

Coordination: The summit workshop leaders reviewed the content of this report for accuracy.

Summit Contributors (in alphabetical order):

- Aberdeen Test Center, U.S. Army
- ABS Technology Architects
- Aeronautical Systems Center, U.S. Air Force
- Aerospace Corporation, The
- Air Force Institute of Technology, U.S. Air Force
- Alion Science and Technology
- Assistant Secretary of Defense for Networks and Information Integration
- Assistant Secretary of the Navy for Research, Development and Acquisition
- ATK–Advanced Weapons Division
- BAE Systems Corporation Armament Systems Division
- Battelle
- C3I Systems Corporation
- Carnegie Mellon University Software Engineering Institute
- Communications-Electronics Command (SEC SED), U.S. Army
- Computer Sciences Corporation
- Center for Strategic and International Studies
- Data Systems Analysts, Inc.
- DDR&E, OSD
- Defense Acquisition University
- Defense Contract Management Agency
- Defense Information Systems Agency
- Defense Systems Management College
- Defense Threat Research Agency
- Fraunhofer Center for Experimental Software Engineering, Maryland
- Harris Corporation
- High Performance Technologies, Inc.
- HQDA, OASA (ALT)
- Institute for Defense Analysis
- ITT Corporation–AES
- Lockheed Martin Corporation, Integrated Systems and Solutions
- Massachusetts Institute of Technology
- MITRE Corporation
- Modern Technology Solutions, Inc.
- National Defense University
- NAVAIR, U.S. Navy
- Naval Academy, U.S.
- Naval Ordnance Safety and Security Activity, U.S. Navy
- Northrop Grumman Corporation
- NUWC Division, Newport, U.S. Navy
- Office of the Assistant Secretary of Defense (C3T)
- ODUSD(A&T) Software Engineering and System Assurance
- Office of Naval Research, U.S. Navy
- Program Executive Office C3T
- Radiant Blue
- RDA CHENG, U.S. Navy
- SAF/AQRE, U.S. Air Force
- Science Applications International Corporation
- Software Technology Support Center
- Space and Missile Systems Center, U.S. Air Force
- Strategic Insight, Ltd.
- Systems and Software Consortium, Inc.
- Unified Cryptologic Architecture

Summit Report Contributors (in alphabetical order):

- Bruce Amato
- Kristen Baldwin
- Robert D. Buckstad
- Grady Campbell
- James Clausen
- Paul R. Croll
- Kathryn Duncan
- Jeffrey Dutton
- Scott Lucero
- Mike Nicol
- Ken Nidiffer
- Lawrence T. Osiecki
- George Prosnik

Final Draft Published: February 26, 2007

Nothing follows