DoD Software Engineering and System Assurance

Focusing on Top Software Issues

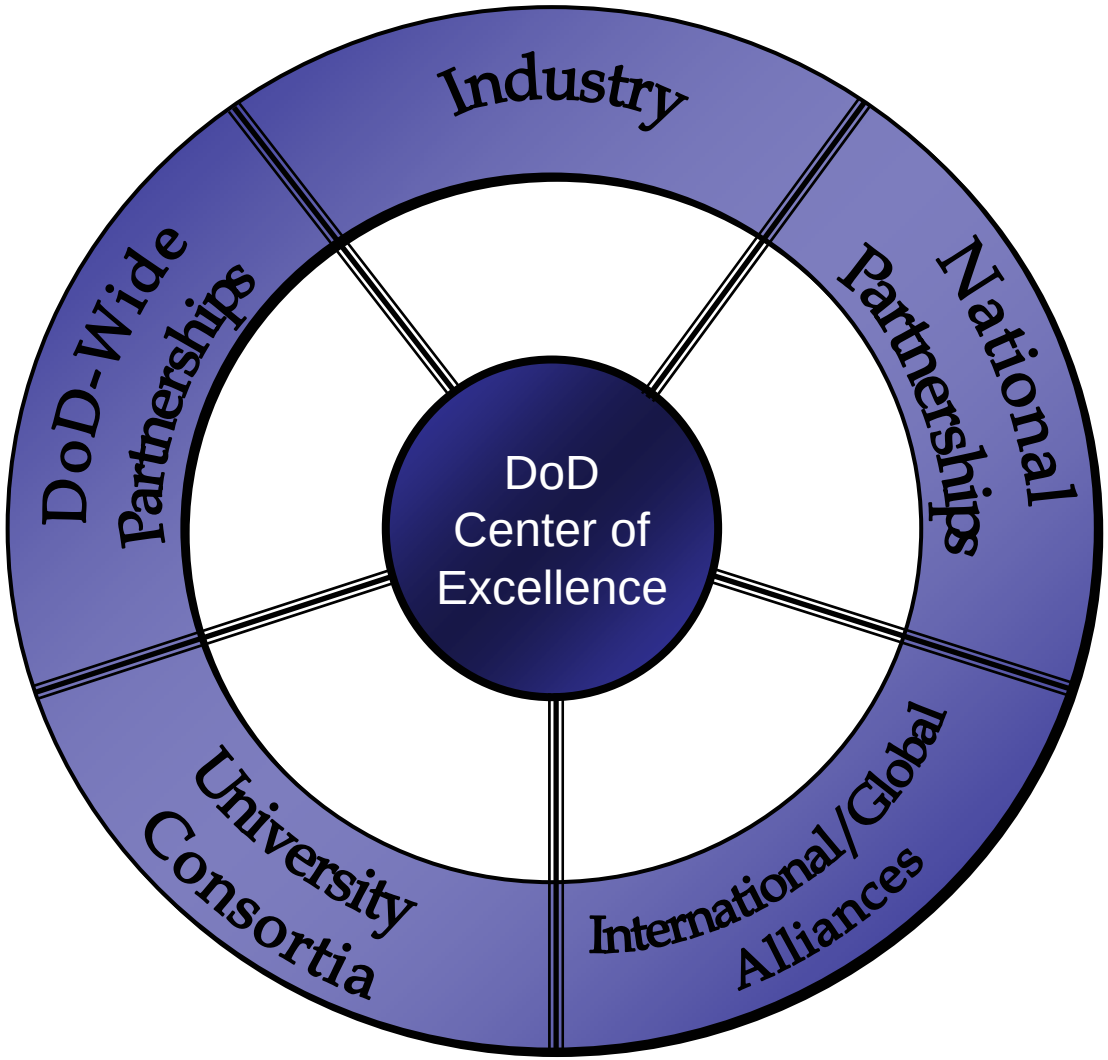
Kristen Baldwin

Deputy Director, Software Engineering and System Assurance

Office of the Under Secretary of Defense

Acquisition, Technology and Logistics

Establishing a DoD Engineering Center of Excellence



OSD Software Engineering and System Assurance

- Support Acquisition Success
- Improve State-of-the-Practice of Engineering
- Leadership, Outreach and Advocacy
- Foster Resources to Meet DoD Needs

Recap of Activities Since Last NDIA Conference

- Documented software issues, needs
  - Software Industrial Base Study completed
  - NDIA Top Software Issues Workshop Report
  - Defense Software Strategy Summit Report
- Created partnerships
  - Established network of DoD software POCs
  - Chartered the NDIA Software Committee and Expert Panel
  - Bi-weekly software collaboration exchanges with Government, Academia, and Industry
  - Restructured the US-UK-AUS Trilateral Working Group
- Performed gap analysis
  - Identified ongoing software initiatives; mapped them to issue areas
  - Two outcomes:
    1. Identified initiatives that deserve cross-DoD attention
    2. Identified gaps where attention is needed

Common Goal: Provide visibility to key initiatives; Focus attention on gaps

SW Issue/GAP Workshop Findings

March 2007

*based on NDIA Top SW Issues, OSD Program Support Reviews, and DoD Software Summit findings

Primary Software Focus Groups*

Software Acquisition Management

- Standards – O, N
- DAG Ch 4/7 – O, AF
- Prog Spt – O, All
- Contract Language – A, M, N
- SW Estimation – GAP
- Lifecycle Policy – AF
- Risk Identification - GAP

Software Development Techniques

- Agile – O, SEI
- Architecture – A, SEI
- COTS – SEI
- Open Source – AF
- Sustainment – GAP
- SW Interoperability – GAP
- SW Test - GAP

SW & SE Integration

- Requirements – GAP
- SE/SW Process Int – O
- SW Council – N
- SW Dev Plan – N
- SW in SEP – N
- SW in Tech Reviews – N
- SW Quality Attributes - GAP

Knowledge Sharing

- Standards – O, N
- DAG Ch 4/7 – O, AF
- Prog Spt – O, All
- Contract Language – A, M, N
- Estimation – GAP
- Lifecycle Policy – AF
- Risk Identification - GAP

Data and Metrics

- SW Metrics – A, O
- SW Cost – O
- SW EVM – DCMA
- SW Estimation - GAP

Human Capital

- Education Sources – N, A
- Leadership Training – A, SEI
- SETA Quals – GAP
- SW Human Cap Strategy – GAP
- Industrial Base – O
- University Curriculum – O
- Workforce Survey - AF

Ongoing Initiative Owners

- O – OSD/SSA
- A – Army
- N – Navy
- AF – Air Force
- M – MDA
- SEI
- DCMA
- GAP – No activity

Ongoing SW Initiatives (w/owners) and Gaps binned to Focus Groups

Identified Software Gap Areas

- Estimation
- Risk Identification
- Sustainment
- Interoperability
- Test
- Requirements
- Quality Attributes
- Qualifications for Software Support (SETA)
- Human Capital Strategy

<u>Needed step:</u>
Develop plans to define, and begin to address, these gaps

Software In Acquisition Workshop October 16-17, 2007

- Purpose: Off-year workshop (Summit held every 2yrs)
  - Share progress on initiatives against known issues
  - Collaborate on gaps
- Format:
  - Leadership updates
  - Presentations from the community to share progress, experiences
  - Workshops to develop action plans for resolution of key issues
    - Requirements, Risk/Cost Estimation, SW Quality Attributes
- Audience:
  - DoD programs, practitioners, industry, FFRDC, academia
- Community forum focused on software in acquisition

Software Requirements Workshop

Requirements Issues

- Many problems are the same as those we have been trying to fix for 30 years
- We know what to do however we do not incentivize it, pay for it and train for it (e.g. application of IEEE requirements attributes)
- Lack of early and continuous involvement of all relevant SMEs
- We are making changes to software whose existing behavior we do not understand
- Do not have adequate tools, methods and processes for requirements definition
- How are articulated specifications adequate to development derived from capabilities? – especially in a continuous evolution environment
- Requirements are added, changed or deleted without sufficient engineering
- Component cannot operate alone – no one owns the external relationships

Requirements Workshop Recommendations

1. Define an effective “software portfolio” management framework
   - Protect the continuity of systems/software and requirement engineering throughout the software life cycle
2. Implement the techniques we know will work and identify any shortcomings
   - Training
   - Incentives
   - Re-examine IEEE tenets for good requirements
3. Find ways to leverage the malleability of software
   - We need new methods to deal with “on the fly” and/or external requirements: Software has the ability to adapt faster than other elements
   - Build and integrate effective modeling of existing systems and addition of new requirements
   - Identify resources and methods to facilitate planning for extended use
   - Find ways to manage the malleability to minimize risk

Requirements Workshop Recommendations

4. Change our view/perspective of “sustainment” to “continuous evolution”
   - Codify the processes for “reverse engineering” candidates to extract for reuse – system components from government or industry
   - Look at organizational as well as methods and skills to perform continuous evolution
5. Establish a research program
   - Identify the characteristics of requirements engineering in type III systems and how it is distinguished from type I
   - Start to identify good practices

Software Risk/Cost Estimation Workshop

Panel #2: Software Estimation / Risk – Top Level Summary

- **Panel Theme/Focus:**
  - Policy, guidance, and training for improving software acquisition and program execution through synergistic integration of risk management and estimation approaches
- **2006 DoD Software Summit Findings:**
  - Must understand that software is a primary performance, schedule, and cost driver
  - Pressure to rapidly procure new capabilities can inhibit balance of life cycle cost, schedule, and performance expectations to achieve executable programs
  - Software risks and life cycle costs are not consistently accommodated in planning
  - Realistic schedule and effort or cost estimates are often rejected or constrained
  - Reuse, open source, and government off-the-shelf software estimating methods are inadequate

Software Estimation/Risk Recommendations (1 of 2)

- Investigate principles for organizing WBS-related artifacts that address Software Engineering sufficiently
  - Concept definition 6 mo, Proposed Language 12 mo, Revised WBS for Mil-Hdbk-881 and related documents

- Investigate strategies and approaches for developing and evolving an integrated software data repository and related tools to address a broad set of stakeholders (government, industry, academia)
  - Concept definition and definition of measures 6 mo, data collection from sample programs 12 mo, Concept of Operations/Business Plan for wide-scale rollout 18 mo.

- Conduct Root Cause analysis studies to understand the problems in software estimation and the use of estimates in the acquisition process
  - Data gathering of lessons learned and studies 6 mo, Draft analysis 12 mo, Prioritization of high leverage areas for improvement 18 mo

Software Risk/Estimation Recommendations (2 of 2)

- Develop and implement an incremental acquisition approach (as well as the overall acquisition framework) that accommodates the uncertainty associated with early software estimates and allows for adjustment and refinement over time
  - Data gathering 6 mo., analysis of data 12 mo., proposed changes to DoD policy and guidance documents 18 mo.

- Establish policy, related guidance, and recommended implementation approaches for software data collection and analysis across all DoD acquisition programs
  - Concept definition 6 mo, initial analysis of data 12 mo, proposed changes to DoD policy and guidance documents 18 mo

Software Quality Attributes Workshop

Software Quality Attribute Priority Recommendations (1 of 2)

1. **Product: Recommended guidance on engineering issues such as: quantitatively identifying, predicting, evaluating, verifying, and validating Quality Attributes**

Actions to include:

1.1. Address tie-in to KPPs and TPMs

1.2. Identify methods for quantitative assessment of individual and integrated Quality Attributes

1.3. Define the specific pieces of evidence required to pass acquisition milestones

1.4. Identify methods for predicting quality attribute outcomes for the delivered system, throughout the life cycle

2. **Product: Recommendations for improving OSD/Service-level acquisition policy regarding Quality Attributes**

Actions to include:

2.1. Identify benefits of addressing software quality attributes as part of an acquisition risk reduction strategy

2.2. Address gaps in SEP, TEMP, JCIDS, DAG

2.3. Develop model RFP language

2.4. Define expectations for Quality Attribute review during Acquisition Milestone Reviews (e.g. PDR)

Software Quality Attribute Priority Recommendations (2 of 2)

3. Product: Taxonomy of software quality attributes and how they are related

Actions to include:

3.1. Collect and organize definitions of Quality Attributes

3.2. Enumerate relationships to systems quality expectations

3.3. Survey existing information on selection and prioritization of software quality attributes for different classes of programs

4. Product: Program Manager guidance on Introduction to Software Architectural Evaluation of Quality Attributes

Actions to include:

4.1. Evaluate existing guidance documents

4.2. Synthesize results into recommended guidance

5. Product: Collaboration site for collecting data, sharing work products, facilitating on-going discussion

Actions to include:

5.1. Identify host/collaboration tool

5.2. Define site framework/rules

2nd Tier Work Products

1. **Product: Catalog of architectural approaches with respect to their Quality Attribute profiles**

Actions to include:

1.1. Develop catalog format and approach

2. **Product: Process for selecting the subset of Quality Attributes for specific systems of interest**

Actions to include:

2.1. Develop strategy for attribute trade-offs

2.2. Identify risk implications

2.3. Develop a checklist of questions to identify attributes important to the stakeholder(s)

3. **Product: Recommendations for basic research on quality attributes**

Actions to include:

3.1. Address inadequacies in state of the art/state of the practice

4. **Product: Guidance on how to engineer quality attributes into systems**

Actions to include:

4.1. Define engineering processes to achieve specific quality attribute levels

4.2. Report on current research and practice

3rd Tier Work Products (1 of 2)

1. **Product: Guidance on addressing Quality Attributes in the Systems of Systems Context**

Actions to include:

1.1. Perform mission thread analysis

1.1.1. Use workshop outline (funded)

1.2. Define systems of Systems software architecture evaluation approach

2. **Product: Examination of Root Cause Analysis Workshop data with respect to Quality Attributes implications**

Actions to include:

2.1. Examine root cause analysis workshop data to determine quality attribute implication

3. **Product: Examination of what DAU teaches regarding Quality Attributes and recommendations for improvement (tied to policy and guidance in #2)**

Actions to include:

3.1. Review course material used for PMs and Systems Engineers about quality attributes

3.1.1. Provide recommendations for additions to course materials

4. **Product: White paper on how to reason about Quality Attributes in architecture model standards (e.g. DODAF)**

Actions to include:

4.1. Produce white paper on how to reason about Quality Attributes in architecture model standards (e.g. DODAF)

3rd Tier Work Products (2 of 2)

5. Product: Guidance on addressing quality attributes of COTS/NDI

Actions to include:

5.1. Develop guidance on addressing quality attributes of COTS/NDI

6. Product: White paper on quality attributes implications of agile methods for large scale defense systems

Actions to include:

6.1. Develop White paper on quality attributes implications of agile methods for large scale defense systems

7. Product: Guidance/lessons learned from commercial practice

Actions to include:

7.1. Collect and provide guidance/lessons learned from commercial practice

OUSD(AT&L)/SSA FOCUSED INITIATIVES

System Assurance Guidebook Project Description

Issue: Systems are vulnerable to malicious tampering

- Project Description:
  - Provide *practical guidance* on augmenting systems engineering practice for system assurance
  - Synthesize existing knowledge from organizations, standards and best practices
  - Recap concepts from standards
- Opportunity for:
  - Practitioners, academe who implement systems engineering, assurance, safety, security, program protection, etc. into processes and programs
- The project addresses
  - Integration of assurance guidance and practices into systems engineering
- Product:
  - Guidebook on Engineering for System Assurance
- Outcome Goal:
  - Intent is to yield assured program / system with demonstrable evidence of assurance

System of Systems Project Description

Issue: No common definition, or guidance for SoS

- Project Description:
  - Effort led by the Office of the Secretary of Defense
  - Collaborative Approach with DoD, Industry, Academia
- Purpose
  - 6 month effort addressing areas of agreement across the community
  - Focus on technical aspects of SE applicable across SoS management constructs
  - Vehicle to capture and debate current SoS experience
- Audience
  - Program Managers and Lead/Chief Engineers
- The project addresses
  - Considerations for engineering above a system level
- Product:
  - SoS Engineering Guide, v1.0, Fall 2007
- Outcome Goal:
  - Program managers/chief engineers have requisite knowledge to manage SoS

SW Engineering Graduate Curriculum Project Description

Issue: There is no commonly accepted structure or content for graduate software engineering education

- Project Description:
  - Develop a core curriculum and core competencies for software engineering
- Opportunity:
  - Industrial and government workforce customers of SWE graduate education
  - Academics who provide SWE and SE graduate education
  - Professional societies with a vested interest in SWE and SE graduate education
- The project addresses
  - Inconsistencies in software graduate degrees
  - Poor definition of labor categories and software expertise
  - The divide between systems and software engineers in industry, government, and academia
  - The project will integrate SE principles and practices into a SWE curriculum.
- Product:
  - An approved curric that can be adopted by the community (industry, academia, associations)
- Outcome Goal(s):
  - Software engineers have a more consistent training base

DoD Acquisition Workforce Software Education Project Description

Issue: DAWIA Curriculum does not address software acquisition issues

- Project Description:
  - Compare identified software competencies with current curriculum
  - Update DAU software acquisition management courseware and other career field training to meet competency needs
  - Develop continuous learning modules as part of the DAU Core Plus construct
  - Initial focus on PM and SPRDE career fields
- Opportunity for:
  - Software and career field process owners and practitioners
- The project addresses
  - Methodology for determining software competencies
  - Methodology for developing tailored solutions for each career field
- Product:
  - Updated DAWIA software competencies reflecting latest policies and guidance
- Outcome Goal:
  - Acquisition professionals have requisite software knowledge

SE/SW Process Integration Project Description

Issue: SE and SW have not been well integrated on projects

- Project Description:
  - Study SE and SW processes, capture ongoing harmonization efforts
  - Assess current guidance
  - Identify opportunities for better integration
- Opportunity for:
  - SE and SW process owners or practitioners
  - Academe who teach/study SE and SW
- The project addresses
  - Integration of SW with requirements, risk management and other SE technical and management processes
- Product:
  - Report and recommendations for SW policy, guidance, and tools to better integrate with SE and Acquisition
- Outcome Goal:
  - Software is a major factor in engineering design and acquisition management decisions

CMMI-Acquisition Project Description

Issue: Acquirers lack an appraisable model for acquisition PI

- Project Description:
  - Using GM CMMI for Outsourcing; pilot and generate CMMI-ACQ
  - Involve broad set of acquisition stakeholders to ensure wide application
- Opportunity for:
  - Process improvement stakeholders
  - Acquiring organizations
- The project addresses
  - Identification of key acquirer activities and products
  - Amplification of CMMI core practices to capture acquirer considerations
- Product:
  - CMMI model for Acquisition (built on CMMI foundation for consistency with CMMI-DEV)
- Outcome Goal:
  - Acquisition organizations implement best practices, and institute organizational process improvement

Additional areas for collaboration…and attention

- Additional projects we are looking into:
  - Software earned value guidance
  - Software metrics
  - Software knowledge portal
- Some key gaps remaining:
  - Software sustainment
  - Software test

Our Challenge

- Given the shortage of software resources and critical software reliance
  - We cannot afford to be stovepiped
  - We must integrate across cross-functional perspectives to improve our software capability
- We must focus on long standing software issues
  - Leverage ongoing activities to make a difference
  - Invest in collaborative efforts where there are gaps
- Now...
  - Work together to address software issues
  - Join the DoD SIA Collaborators – participate in bi-weekly collaboration telecons
  - Contribute to workshop action items, and/or ongoing initiatives
  - Contact us at ATL-SSA@osd.mil

Become a DoD Center of Excellence

Backup Slides

Elements of a DoD Strategy for Software

- Support Acquisition Success
  - Ensure effective and efficient software solutions across the acquisition spectrum of systems, SoS and capability portfolios
- Improve the State-of-the-Practice of Software Engineering
  - Advocate and lead software initiatives to improve the state-of-the-practices through transition of tools, techniques, etc.
- Leadership, Outreach and Advocacy
  - Implement at Department and National levels, a strategic plan for meeting Defense software requirements
- Foster Software Resources to meet DoD needs
  - Enable the US and global capability to meet Department software needs, in an assured and responsive manner

Promote World-Class Leadership for Defense Software Engineering

Next Steps

Near Term:

- Determine metrics for each of the 6 Focus Areas
  - Based upon source reports (ie. SW Summit, Top Issues, PSRs, Historical SW Studies)
- Coordinate ongoing initiatives (via Working Group Participation, Defense Software in Acquisition Collaborators)
  - Support and/or leverage initiatives where appropriate
  - Provide visibility across the Department
- Determine action plans for each gap considering:
  - Priority
  - Near Term/Long Term impacts
  - NDIA SW Committee, others, interest in accepting gap(s)
- Engage other communities and participants
  - IT, Business, Research

Over Time:

- Reassess ongoing initiatives against focus area metrics
  - Determine new gaps, or additional effort required to address core issues
- Reassess focus area metrics against systemic software issues
  - From future SW Summits, Systemic Analysis, etc...

*Top Software Issues**

1. The impact of requirements upon software is not consistently quantified and managed in development or sustainment.
2. Fundamental system engineering decisions are made without full participation of software engineering.
3. Software life-cycle planning and management by acquirers and suppliers is ineffective.
4. The quantity and quality of software engineering expertise is insufficient to meet the demands of government and the defense industry.
5. Traditional software verification techniques are costly and ineffective for dealing with the scale and complexity of modern systems.
6. There is a failure to assure correct, predictable, safe, secure execution of complex software in distributed environments.
7. Inadequate attention is given to total lifecycle issues for COTS/NDI impacts on lifecycle cost and risk.

***NDIA Top Software Issues Workshop August 2006**

DoD Software -- What We're Seeing*

- Software systemic issues are significant contributors to poor program execution
  - Software requirements not well defined, traceable, testable
  - Immature architectures, COTS integration, interoperability, obsolescence (electronics/hardware refresh)
  - Software development processes not institutionalized, planning documents missing or incomplete, reuse strategies inconsistent
  - Software test/evaluation lacking rigor and breadth
  - Schedule realism (compressed, overlapping)
  - Lessons learned not incorporated into successive builds
  - Software risks/metrics not well defined, managed

*Based on ~65 program reviews to date