Top Software Engineering Issues in the Defense Industry

NDIA Systems Engineering Division

and Software Committee

September 26, 2006

Task Description

Identify Top 5 Software Engineering problems or issues prevalent within the defense industry

- NDIA Workshop August 24-25, 2006, Washington D.C.

Document issues

- Description and current state
- Rationale and SW impacts
- Develop recommendations (short term and long term)

Generate task report

- Submit to OSD

Task Group Participants

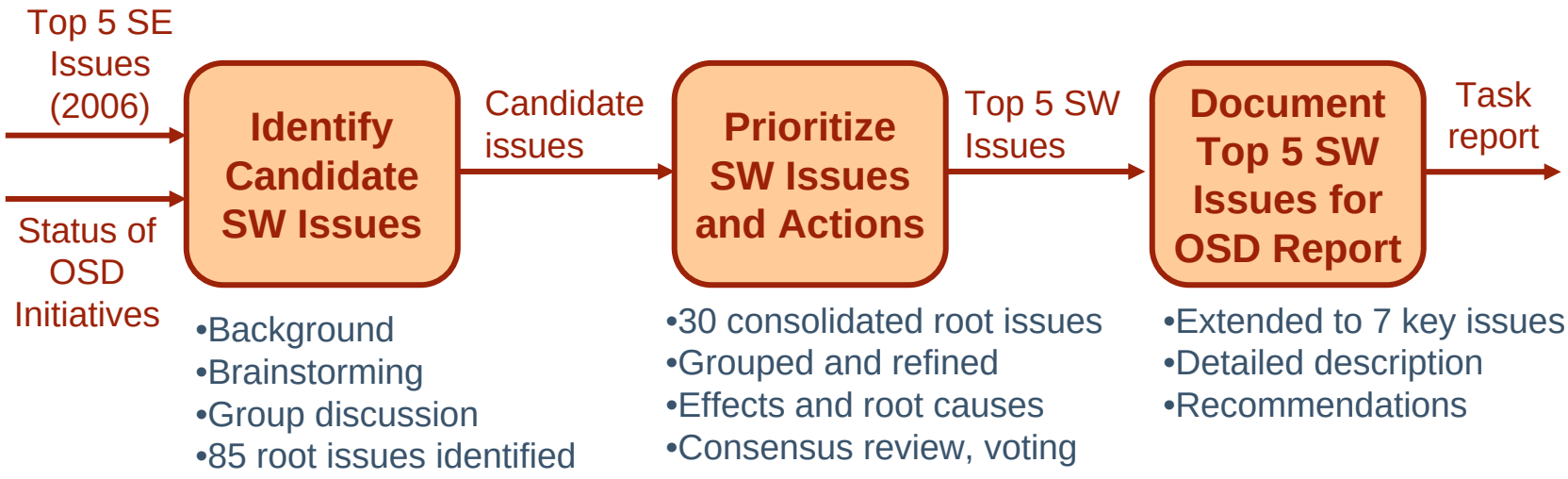
26 workshop participants from industry, government, academia

- **Additional inputs received via email**

BAE Systems
The Boeing Company
Center for Strategic and Intl. Studies
Computer Sciences Corporation
Department of Defense (DoD)
Defense Contract Mgmt. Agency (DCMA)
Florida Institute of Technology
Harris Corporation
Jacobs Technology
Lockheed Martin Corporation
Massachusetts Institute of Technology (MIT)
The MITRE Corporation
Naval Air Systems Command
OUSD (A&T)
Open Systems Joint Task Force
Raytheon Company
RDECOM ARDEC
Asst. Secy. of the Air Force (Acq), SE Division
Software Engineering Institute (SEI)
Systems and Software Consortium
U.S. Air Force Aeronautical Systems Center

Task Group Leads:
Geoff Draper (Harris)
Jeff Dutton (Jacobs Technology)

Task Group Activities



Common Themes Noted

Increasing demands for rapid response to changing threats

- New warfighter capabilities
- Unprecedented complexity
- Increasing dependence on large, software-intensive systems

Evolving acquisition environment

- Innovative and aggressive acquisition strategies
- Extensive reuse of legacy components and system portfolios
- Reductions in acquisition work force due to force-shaping measures

Current approaches for acquiring, developing, verifying, and sustaining software enabled systems are inadequate to deal with the complexities of a dynamic and changing acquisition environment.

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 domain-knowledgeable 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.

Issue 1: The impact of system requirements upon software is not consistently quantified and managed in development or sustainment.

Discussion Points:

- Impacts of changes to requirements not addressed consistently.
- SW requirements are not always adequately addressed in system level requirements or solicitations/contracts.
- Conditions leading to SW reqts changes are not clearly understood.
- Weak linkage to SW requirements for capabilities, portfolios.

Recommendations:

Enforce effective software requirements development and management practices, including assessment of change impacts, for both the acquirer and supplier organizations.

- Strengthen policies/guidance for full requirements traceability.
- Evaluate effectiveness of models for estimating SW and changes.
- Improve involvement of SW and acquisition expertise
  - Early requirements definition (JCIDS)
  - Balance user requirements with cost/schedule constraints
  - Impact assessment and communication of requirement changes

Issue 2: Fundamental system engineering decisions are made without full participation of software engineering.

Discussion Points:

- SW not consistently involved in architectural decisions early in the life cycle.
- Must have authority to participate in system-level decision making and trades.
- SE and SW life cycles, processes, and products are not always consistent or harmonized for meaningful cross-discipline integration to occur.
- Segregation of SE/SW proposal inputs can impede coordination.
- System development methods do not properly leverage SW ability to rapidly field enhanced capabilities.

Recommendations:

Institutionalize the integration and participation of software engineering in all system engineering activities.

- Designate an empowered SW architect (acquirer and supplier) early.
- Update policies, standards, guidance to emphasize SW in decision-making.
- Promote effective SE/SW communications and cross-training.
- Adopt acquisition strategies and development approaches that leverage rapid SW development cycles (iterative capability, sustainability).

Issue 3: Software life cycle planning and management by acquirers and suppliers is ineffective.

Discussion Points:

- SW risks and life cycle costs are not consistently accommodated in planning.
- Realistic schedule and effort (cost) estimates are often rejected or constrained.
- Pressure to rapidly procure new capabilities can inhibit balance of life cycle cost, schedule, performance expectations to achieve executable programs.
- Lack of SW expertise applied in planning and management.
- SW processes/methods improperly selected, tailored, executed, monitored.
- SW measures are not used effectively or acted upon.

Recommendations:

Establish a culture of quantitative planning and management, using proven processes with collaborative decision-making across the software life cycle.

- Establish collaborative contractual relationships enabling joint ownership of issues.
- Change culture to base expectations on historical data and realistic constraints.
- Emphasize selection of appropriate processes/methods and codify in contract.
- Incentivize life-cycle perspective and iteration to circumvent near-term thinking.

Issue 4: The quantity and quality of domain-knowledgeable software engineering expertise is insufficient to meet the demands of the government and the defense industry.

Discussion Points:

- Insufficient highly skilled domain-knowledgeable SW professionals to meet the growing demand, due to inadequate funding, insufficient career incentives, competition, downsizing, etc.
- Insufficient infrastructure to rapidly transition skilled staff to new programs.
- Inadequate SW staffing across the system life cycle (architecture, trades, etc.)
- Career path incentives are insufficient to attract SW professionals.

Recommendations:

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.

- Develop a Software Competency Roadmap, with strategic decisions about the future workforce (competencies, downsizing, R&D, etc.)
- Establish joint government, industry, academia working group to identify strategies for expanding SW resource pool and skill sets.
- Cross-train among functional disciplines (PM, SE, etc.) to expand SW engineering pool and extent of domain knowledge.

Issue 5: Traditional software verification techniques are costly and ineffective for dealing with the scale and complexity of modern systems.

Discussion Points:

- Over-reliance on testing alone rather than robust SW verification techniques.
- Manual testing techniques are labor-intensive, scale poorly, and are unproductive relative to the large investment of resources.
- Compliance-based tests do not adequately cover risks or failure conditions.
- Tests are over-documented with disproportionate effort on detailed procedures.
- Education, training, certifications are inadequate to develop effective test skills.

Recommendations:

Study current software verification practices in industry, and develop guidance and training to improve effectiveness in assuring product quality across the life cycle.

- Sponsor a study of state-of-the-practice verification and testing approaches.
- Review/update testing policies and guidance to emphasize robust, productive approaches that maximize ROI.
- Review adequacy of verification plans/approaches early in the acq. life cycle.
- Emphasize skilled investigation throughout the life cycle, based on coverage, risk mitigation, high volume automation.
- Strengthen curricula, training, certifications, career incentives for testing roles.

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

Discussion Points:

- SW is inherently vulnerable to widespread SW assurance threats – we must be confident in the supply chain pedigree.
- Current techniques are inadequate to verify assured components with well-understood properties.
- Assurance of systems, and SoS, cannot be easily inferred from components due to issues such as composability, emergent behavior.
- Exhaustive testing to rule out vulnerabilities is not feasible.

Recommendations:

Collaborate with industry to develop approaches, standards, and tools addressing system assurance issues throughout the acquisition life cycle and supply chain.

- Establish collaborative initiatives to develop and deploy comprehensive system assurance approaches.
  - Resistance to intrusion and compromise
  - Commercial standards to address vulnerability throughout the supply chain
- Define SW assurance quality attributes for addressing in architectural trades.
- Sponsor system assurance research, policy, guidance, training..

Issue 7: Inadequate attention is given to total lifecycle issues for COTS/NDI impacts on lifecycle cost and risk.

Discussion Points:

- Inadequate estimating methods for COTS/NDI (reuse, open source, GOTS, …)
- COTS/NDI best practices are known but not consistently implemented.
- Inadequate attention to sustainment issues early in the life cycle.
- Open source licensing issues can expose organizations to liability, loss of data rights, potentially large rework.
- Customer expectations for customization reduce benefits of COTS solutions.
- Insufficient infrastructure to create and facilitate reuse across organizations.

Recommendations:

Improve and expand guidelines for addressing total lifecycle COTS/NDI issues.

- Encourage adoption of COTS/NDI best practices (SEI, AIAA, etc.)
- Ensure COTS/NDI life cycle processes are addressed in program plans.
- Review COTS/NDI life cycle support issues and tradeoffs at program milestones and reviews.
- Ensure COTS/NDI issues are addressed in OSD policy and SW Assurance initiative.

Summary Recommendations
Top SW Issues 2006

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 assuring 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 lifecycle COTS/NDI issues.