Engineering Practices for System Assurance

NDIA System Assurance Committee

Presented by

Paul R. Croll

Industry Co-Chair

Computer Sciences Corporation
pcroll@csc.com

Outline

- Definition Of The Problem
- Results From Joint Industry/ Government Forums Addressing Issues In System Assurance
- NDIA System Assurance Committee
- Collaboration With Industry Consortia And Standards Bodies
- DoD Handbook For System Assurance

Definition Of The Problem

Background

- In July 2003, the Assistant Secretary of Defense for Networks and Information Integration [ASD(NII)] established a Software Assurance Initiative to examine software assurance issues
- On 23 Dec 04, Undersecretary of Defense for Acquisitions, Technology and Logistics [USD(AT&L)] and ASD(NII) established a Software Assurance (SwA) Tiger Team to:
  - Develop a holistic strategy to reduce SwA risks within 90 days
  - Provide a comprehensive briefing of findings, strategy and plan
- On 28 Mar 05, Tiger Team presented its strategy to USD(AT&L) and ASD(NII) and was subsequently tasked to proceed with 180 day Implementation Phase

The Software Assurance Context

- The importance of software in the functionality of defense systems has increased dramatically both in weapon systems and command/control and information systems.
- The software content of such systems is most often an amalgamation and integration of various software subsystems, from a myriad of sources. Legacy content is often high in upgraded systems.
- Development of that software often spans multiple companies and countries, and often it is difficult to assure the source of software from a national origin perspective

Software Assurance Problem

- ***Scope***: Software is fundamental to the GIG and critical to all weapons, business and support systems
- ***Threat agents***: Nation-states, terrorists, criminals, rogue developers who:
  - Gain control of systems through supply chain opportunities
  - Exploit vulnerabilities remotely
- ***Vulnerabilities***:
  - Intentionally implanted logic (e.g., back doors, logic bombs, spyware)
  - Unintentional vulnerabilities maliciously exploited (e.g., poor quality or fragile code)
- ***Consequences***: The enemy may steal or alter mission critical data; corrupt or deny the function of mission critical platforms

Software Assurance

Software Assurance concerns the level of confidence that software functions as intended and is free of vulnerabilities, either intentionally or unintentionally designed or inserted as part of the software.

DoD SwA Strategy: Primary Elements

- Partner with Industry to focus science and technology on research and development of technologies to:
  - Improve assured software development tools and techniques
  - Strengthen standards for software partitioning and modularity
  - Enhance vulnerability discovery
- Employ repeatable Systems Engineering (SE) and test processes to:
  - Identify, assess, and isolate critical components
  - Mitigate software vulnerabilities
- Leverage and coordinate with industry, academia and national and international partners to address shared elements of the problem

DoD SwA Strategy Guiding Principles

- Understanding problem from a systems perspective
- Response commensurate with risk
- Sensitivity to potential negative impacts
  - Degradation of our ability to use commercial software
  - Decreased responsiveness/ increased time to deploy technology
  - Loss of industry incentive to do business with the Department
  - Minimize burden on acquisition programs
- Exploitation and extension of relationships with:
  - National, international, and industry partners
  - Ongoing assurance initiatives, e.g., trusted integrated circuits and Information Assurance

The Systems Engineering Challenge

Integrating a heterogeneous set of globally engineered and supplied proprietary, open-source, and other software; hardware; and firmware as well as legacy systems to create well-engineered integrated, interoperable, and extendable systems whose security and other risks are acceptable – or at least tolerable.

Systems Assurance

Systems assurance establishes the grounds for a justified level of confidence that the system behaves as intended and specified; as well as establishing the degree of uncertainty involved, for example about the system's vulnerability.

Systems Engineering for SwA - Many Alternatives to Consider

- Design around the problem
  - Added emphasis on DoD systems engineering practices to mitigate COTS-based risks
- Build better products
  - Vector commercial products to enhance bounding and controllability
- Better understanding of what’s in the product
  - Enhance transparency, testability, and understandability of product software code
- Use High Assurance products selectively where needed
  - Use DoD security components in critical functions and at key architecture junctures
- Many more possible avenues…

Results From Industry/ Government Forums

NDIA Software Assurance Summit Summary Findings

- Standards, Metrics, and Models
  - Many techniques, tools, and practices, but no consensus on one set of best practices on this topic
  - No consensus on definition or magnitude of problem
- Industry Best Practices
  - Lack of software assurance knowledge
  - Lack of disciplined application of good software development practices
  - Need to reach out to a broader community to capture ongoing best practices
- Engineering Processes
  - Policy and SOW language to explicitly call out Software Assurance
- Science and Technology
  - Establish relationships with key companies and consortia
  - Participate in key forums and workshops
  - Collaborate with University Centers of Excellence

DHS/DoD SW Assurance Summit Issues and Observations

- We can discuss what we need in software security, but vendors won't do it until we make it worth their while.
  - Need to factor it into the acquisition process
  - How do we incentivize the industry to go out and build secure software?
- Efficiencies and bottom line concerns often do not involve software assurance.
  - No way to put the time and effort in to deliver secure software.
  - If government is not willing to put it in as a hard requirements – you won't get it.
- A study of patch management costs vs. costs of developing secure software might provide some ammunition for doing it right the first time.
- COTS is different – low price comes essentially at an assurance cost
  - Need to understand what you're actually getting for the price, for commodity software
- What are the adverse consequence for vendors when they don't meet security requirements?
  - When we use commercial products we sign a waiver assuming all risk and liability

Defense Software Strategy Summit

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..

Defense Software Strategy Summit

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.

NDIA System Assurance Committee

NDIA System Assurance Committee Mission Statement

Assure effective functionality of our command, control, communications and related weapon systems with high confidence that the systems are not vulnerable to intrusion and cannot be compromised.

NDIA System Assurance Committee Charter

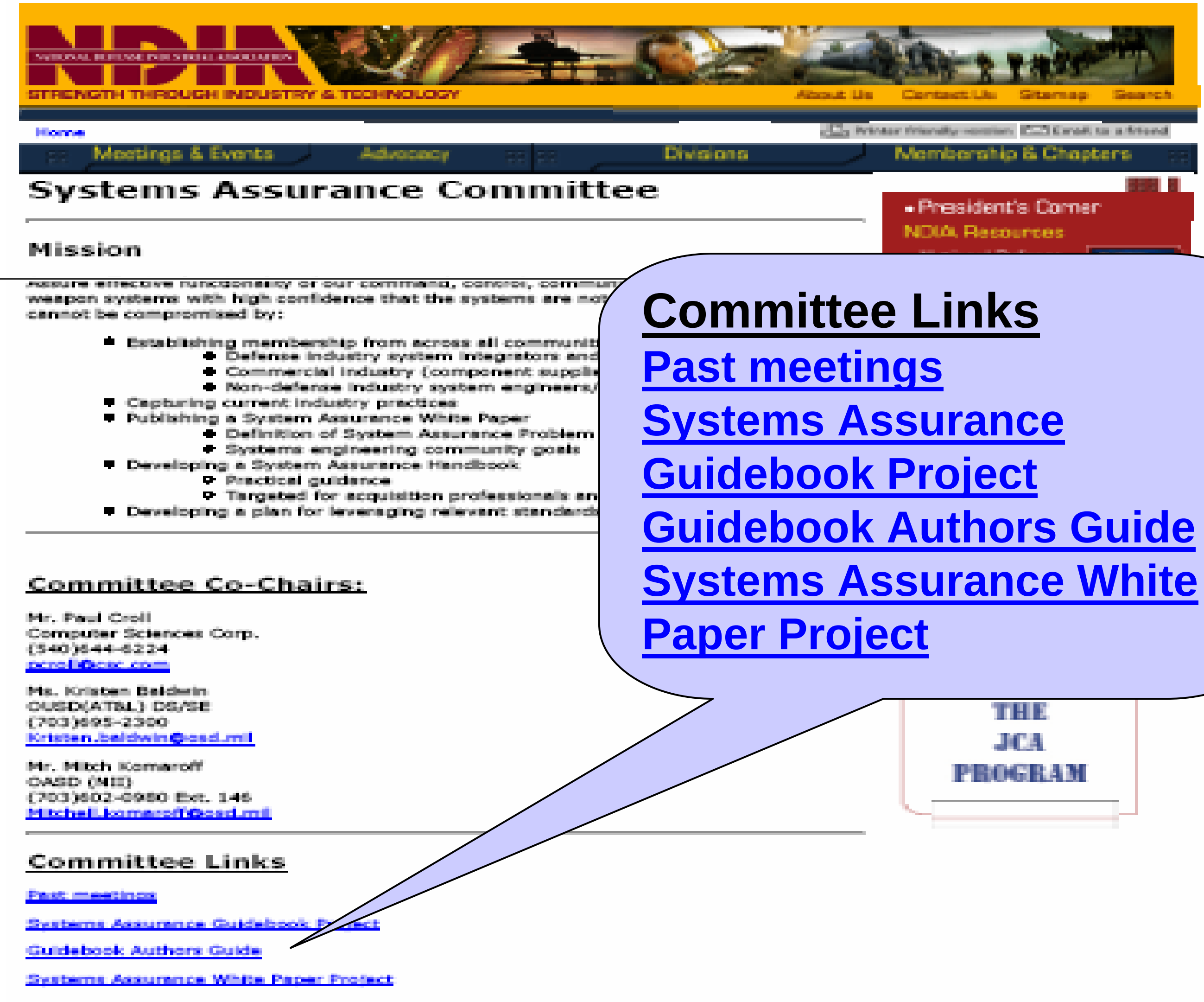
- Create an extended community to engage in system assurance strategy
- Comment on recommendations coming out of DoD strategy
- Leverage standards activities
- Develop a System Assurance Guidebook
- Chairs
  - Paul Croll, NDIA SED
  - Kristen Baldwin, OUSD AT&L
  - Mitchell Komaroff, ASD NII

NDIA System Assurance Committee Outcomes

- Support acquirers (new and existing programs)
  - Enhance guidance and recommended practices
  - provide the tools and techniques to meet assurance requirements
- Procure systems with known assurance properties
- Foster a competitive commercial environment that values software assurance as a product feature

Committee Web Site

www.ndia.org/Content/ContentGroups/Divisions1/Systems_Engineering/Systems_Assurance_Committee.htm



Systems Assurance Committee

Mission

Assure effective functionality of our command, control, communi
weapon systems with high confidence that the systems are not
cannot be compromised by:

- Establishing membership from across all communit
  - Defense industry system integrators and
  - Commercial industry (component supplie
  - Non-defense industry system engineers/
- Capturing current industry practices
- Publishing a System Assurance White Paper
  - Definition of System Assurance Problem
  - Systems engineering community goals
- Developing a System Assurance Handbook
  - Practical guidance
  - Targeted for acquisition professionals an
- Developing a plan for leveraging relevant standard

Committee Co-Chairs:

Mr. Paul Croll
Computer Sciences Corp.
(540)644-6224
pcroll@csc.com

Ms. Kristen Baldwin
OUSD(AT&L) DS/SE
(703)695-2300
Kristen.baldwin@osd.mil

Mr. Mitch Komaroff
OASD (NII)
(703)602-0980 Ext. 146
Mitchell.komaroff@osd.mil

Committee Links

Past meetings

Systems Assurance Guidebook Project

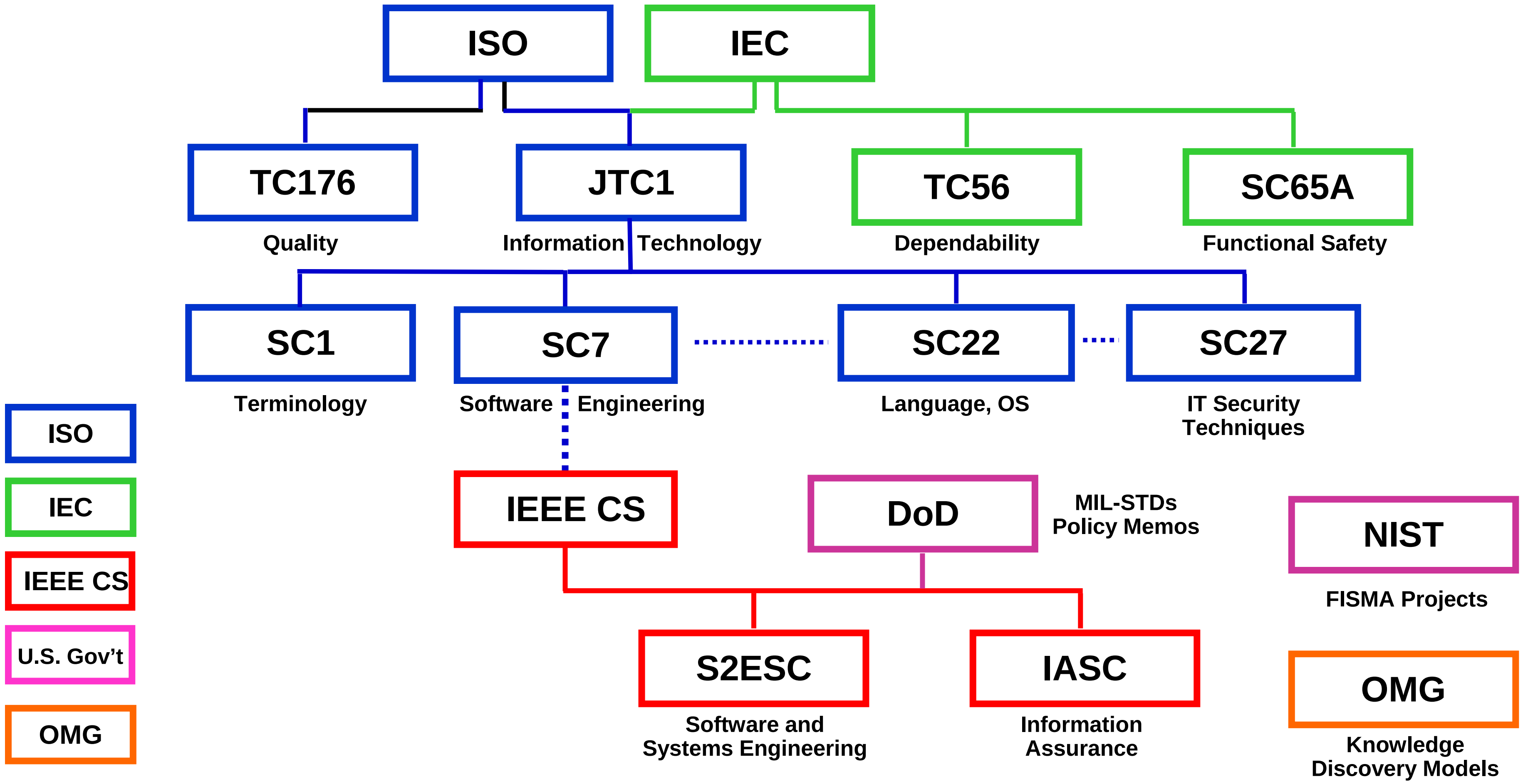
Guidebook Authors Guide

Systems Assurance White Paper Project

Committee Links
Past meetings
Systems Assurance Guidebook Project
Guidebook Authors Guide
Systems Assurance White Paper Project

Collaboration With Industry Consortia And Standards Bodies

Standards Organizations Supporting Assurance

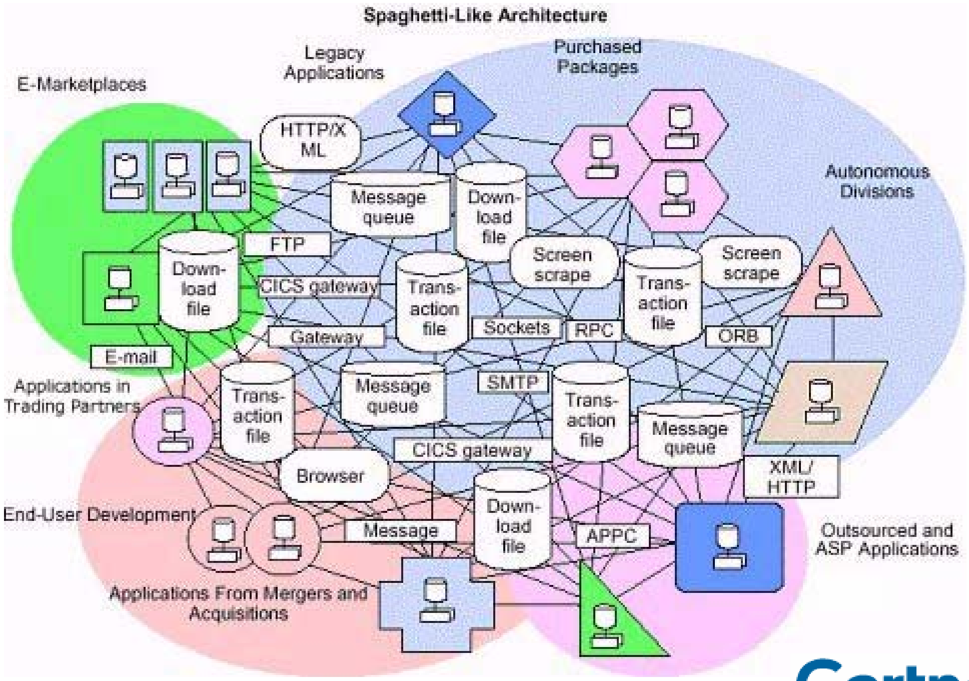


OMG Software Assurance Framework

- Goal
  - – Achieve transparency of the product
- Objective
  - – Provide a standard for a collaborative framework to facilitate information exchange regarding claims, arguments, evidences, consequences, risks
- Mechanism
  - – A layered Knowledge Discovery Model (KDM) to facilitate extraction of artifacts and relationships from source code
    - Focuses on the existing system and how one extracts artifacts to help validate product security

The Knowledge Discovery Problem: *Complexity, Multiple Technologies, Multiple Vendors*

- *Complexity*
- *COTS/GOTS/Reuse*
- *Existing software assets*
- *Hybridization*
- *Evolution*
- *Erosion of Knowledge*
- *Globalization*
- *Lack of Skills*

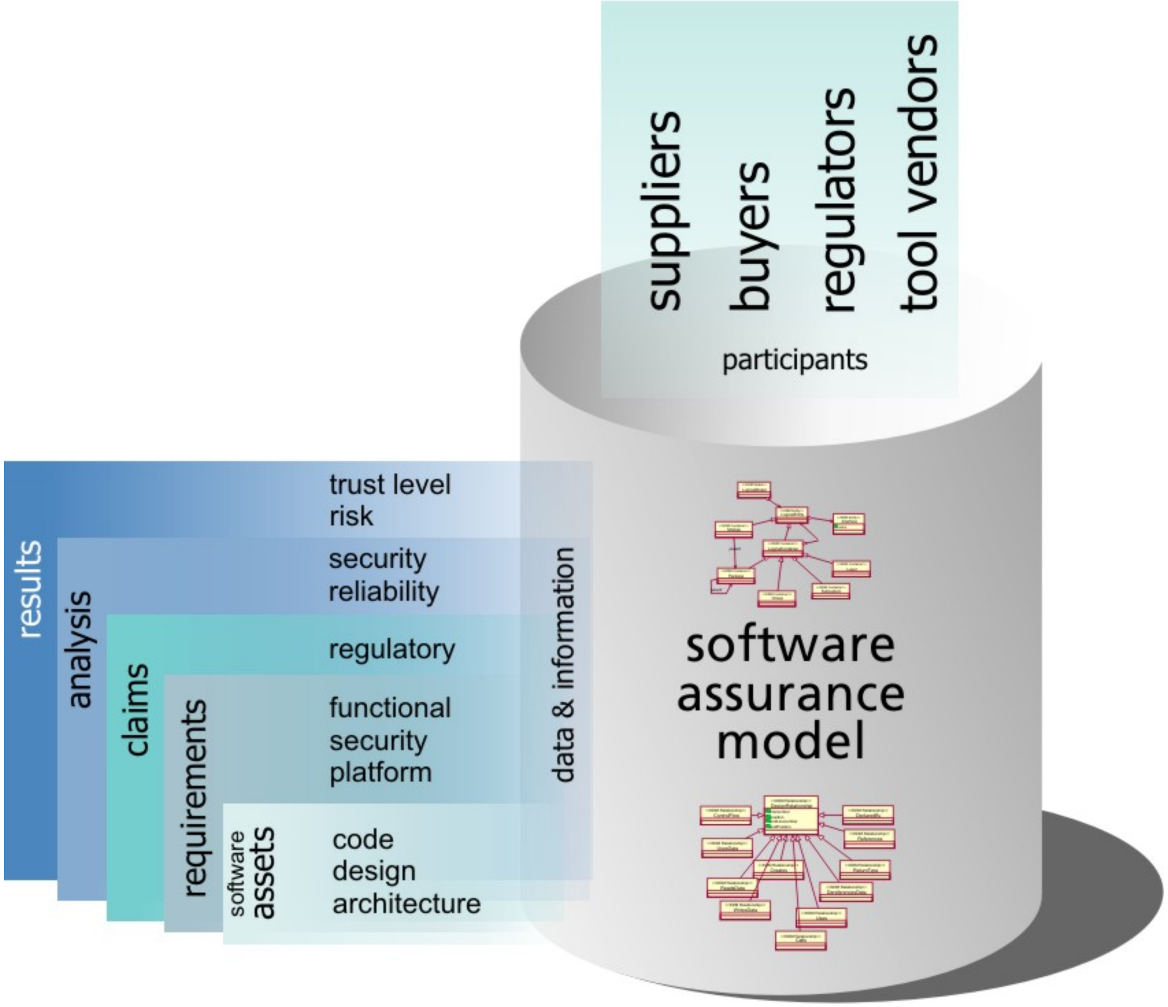


Gartner

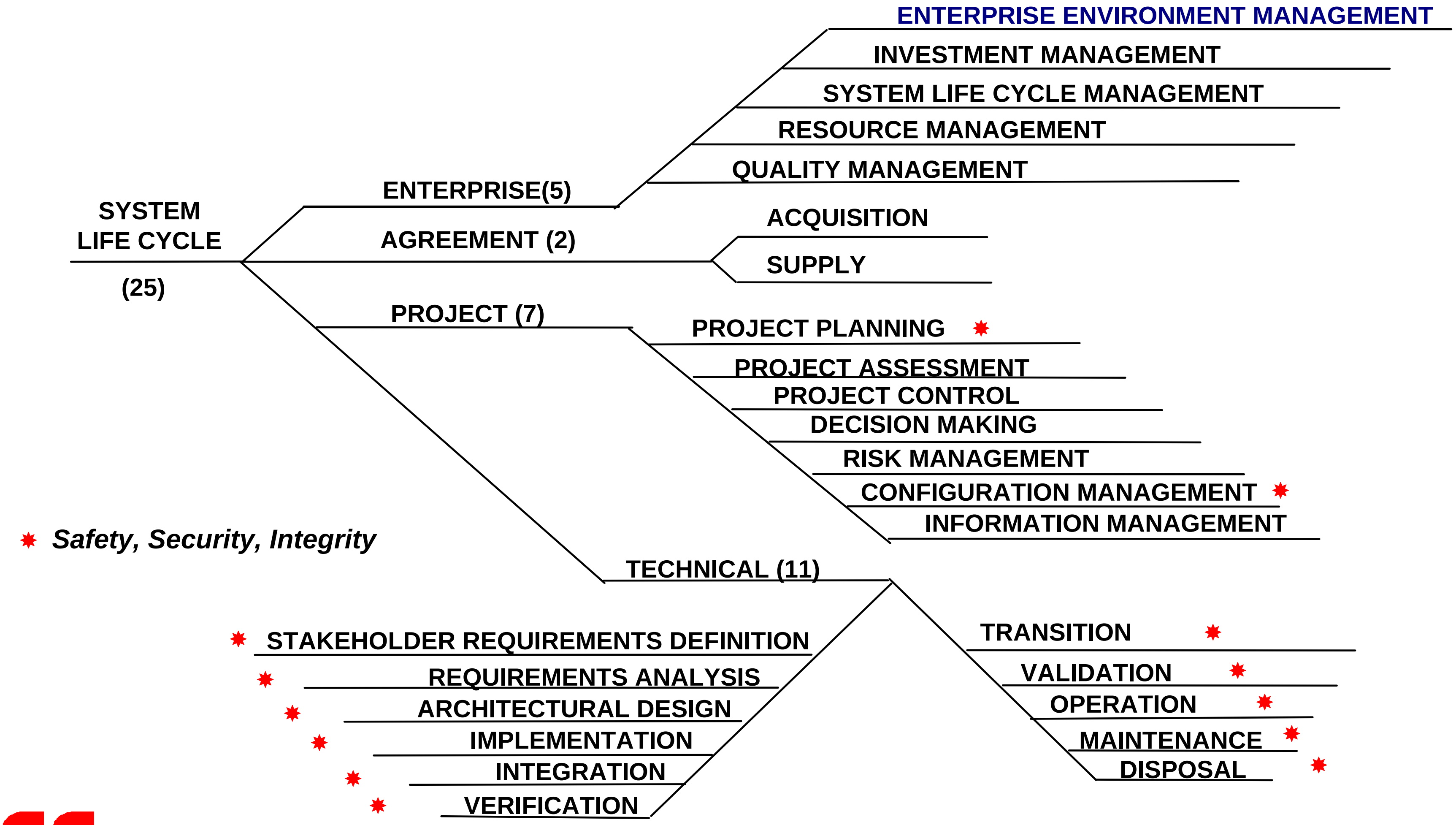
These factors make increasingly difficult to assess the properties of software systems, such as quality, reliability, performance, robustness, and trustworthiness

Source: Djenana Campara
CEO, KDMAnakytics
OMG, Co-Chair: ADMTF & SwA

Focus Is On A Coordinated Model Strategy



Assurance In The ISO/IEC 15288 System Life Cycle Process Framework



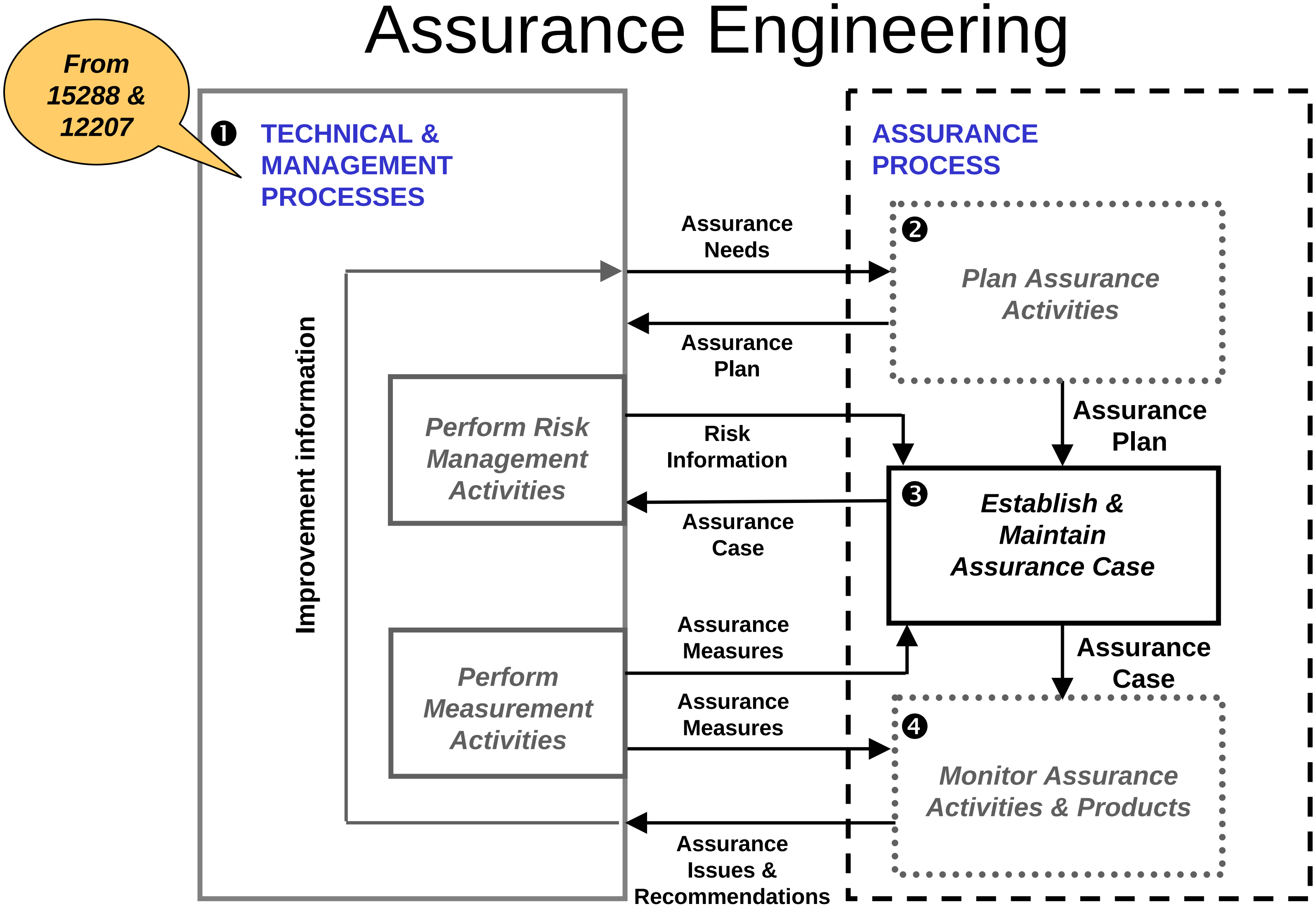
- SYSTEM LIFE CYCLE (25)
  - ENTERPRISE(5)
    - ENTERPRISE ENVIRONMENT MANAGEMENT
    - INVESTMENT MANAGEMENT
    - SYSTEM LIFE CYCLE MANAGEMENT
    - RESOURCE MANAGEMENT
    - QUALITY MANAGEMENT
  - AGREEMENT (2)
    - ACQUISITION
    - SUPPLY
  - PROJECT (7)
    - PROJECT PLANNING ✸
    - PROJECT ASSESSMENT
    - PROJECT CONTROL
    - DECISION MAKING
    - RISK MANAGEMENT
    - CONFIGURATION MANAGEMENT ✸
    - INFORMATION MANAGEMENT
  - TECHNICAL (11)
    - ✸ STAKEHOLDER REQUIREMENTS DEFINITION
    - ✸ REQUIREMENTS ANALYSIS
    - ✸ ARCHITECTURAL DESIGN
    - ✸ IMPLEMENTATION
    - ✸ INTEGRATION
    - ✸ VERIFICATION
    - TRANSITION ✸
    - VALIDATION ✸
    - OPERATION ✸
    - MAINTENANCE ✸
    - DISPOSAL ✸

✸ *Safety, Security, Integrity*

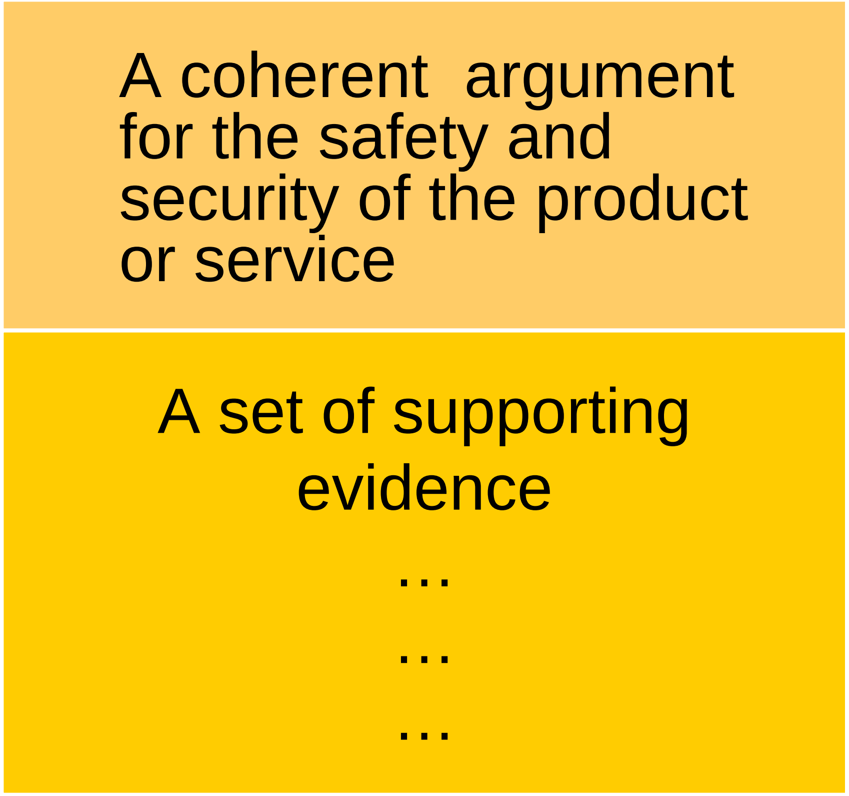
IEEE P15026 – System And Software Assurance Engineering

- Describes an assurance process that provides sufficient evidence that a system satisfies its critical requirements throughout the life cycle of a product or service
- Consists of the following activities:
  - Plan assurance activities
  - Establish and maintain the assurance case
  - Monitor assurance activities and products
- Elaborates on IEEE (ISO/IEC)15288 and 12207

IEEE P15026 – System And Software Assurance Engineering


From 15288 & 12207
❶ TECHNICAL & MANAGEMENT PROCESSES
ASSURANCE PROCESS
Improvement information
Perform Risk Management Activities
Perform Measurement Activities
Assurance Needs
Assurance Plan
Risk Information
Assurance Case
Assurance Measures
Assurance Measures
Assurance Issues & Recommendations
❷ Plan Assurance Activities
Assurance Plan
❸ Establish & Maintain Assurance Case
Assurance Case
❹ Monitor Assurance Activities & Products

The Assurance Case – Structure And Attributes



A coherent argument for the safety and security of the product or service

A set of supporting evidence

…

…

…

Attributes

- Clear
- Consistent
- Complete
- Comprehensible (to all relevant stakeholders)
- Bounded
- Defensible
- Should cover all stages of the life cycle

The Assurance Argument

- A high level summary of the claim(s)
- Justification that the product or service is ***acceptably*** safe, secure, or dependable
- Rationale for claiming a specified level of safety and security
- Citation of relevant standards and regulatory requirements
- Identification of the configuration baseline
- Identified hazards and threats and the residual risk of each hazard and threat
- Operational and support assumptions

ISO SC27 IT Security Standards

- Key Standards
  - ISO/IEC 27000 series – Information Security Management System (ISMS)
  - ISO/IEC 17799:2005 – Code of Practice for Information Security Management
  - ISO/IEC 21827, System Security Engineering Capability Maturity Model (SSE CMM) revision
  - ISO/IEC 15443, FRITSA
    - Part 1: A framework for IT security assurance
    - Part 2: Assurance methods
    - Part 3: Analysis of assurance methods
  - ISO/IEC DTR 19791, Assessment of Operational Systems
- SC27 looking at new topics
  - Privacy
  - Identify management
  - Biometric protection and evaluation

DoD Guidebook For System Assurance

Guidebook Overview

- The intent of the Guidebook is to provide practical guidance for NDIA members, the larger contractor community, academe, and other commercial and government partners.
- It is a synthesis of knowledge gained from existing practice, recommendations, policy, and mandate, rather than a reinvention of anyone’s wheel.
- It recaps important concepts and principles from foundational documents, standards, and mandates, and discuss them in the larger context of systems assurance as presented by the white paper.

Expected Outcome

A set of engineering activities to provide the basis for justified increase the level of confidence that the risk for the system is acceptable, *or at least tolerable*

Guidebook Engineering Practices

- Describe differences required in system engineering because of concerns for:
  - Maliciousness
  - reducing uncertainty
  - providing an explicit basis for justified confidence
- Provides the basis for rational decisions regarding selection, delivery, acceptance, and use
- Addresses the origin and pedigree of systems and their parts
  - both hardware and software
- Limited to the computational components of a system and related needs and consequences
- Consistent with the Defense Acquisition Guide (DAG)

Guidebook to DAG Mapping

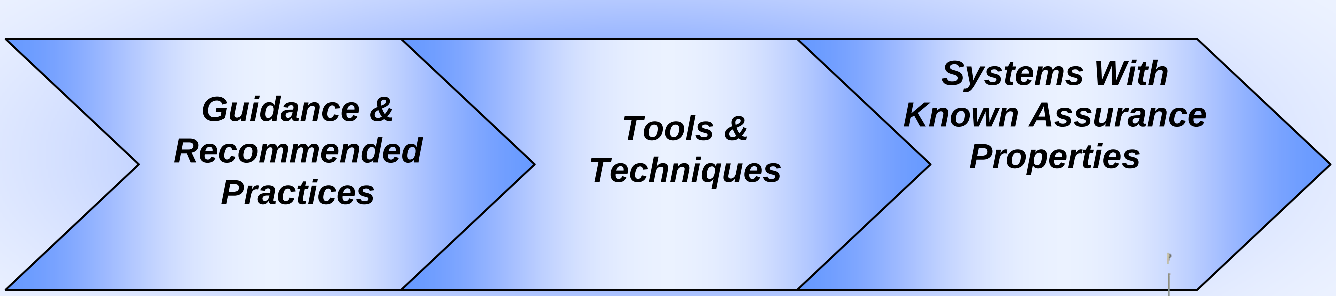
Guidebook (Section 6.4)	DAG (Chap 4)
6.4.2 Stakeholder Requirements Definitions	4.2.4.1 Requirements Development
6.4.3 Requirements Analysis	4.2.4.2 Logical Analysis
6.4.4 Architectural Design	4.2.4.3 Design Solution
6.4.5 Implementation	4.2.4.4 Implementation
6.4.6 Integration	4.2.4.5 Integration
6.4.7 Verification	4.2.4.6 Verification
6.4.8 Transition	4.2.4.8 Transition
6.4.9 Validation	4.2.4.7 Validation
6.4.10 Operation (and training)	4.3.5 Ops & Support
6.4.11 Maintenance	4.3.4 Production & Deployment
6.4.12 Disposal	-.-.-. No Match

Guidebook
High Level Outline

- Front Matter
  - Abstract, Introduction
  - Authors and Reviewers
  - Contacts in Communities of Interest and Practice
  - Correspondence with Existing Standards
  - White Paper content
- Body (ISO/IEC 15288)
- Back Matter
  - Case Examples
  - Documentation templates

The Way Forward

Competitive Commercial Environment



Guidance & Recommended Practices

Tools & Techniques

Systems With Known Assurance Properties

We Welcome Your Participation

Please Join The

NDIA System Assurance Committee

For More Information . . .

Paul R. Croll
Computer Sciences Corporation
5166 Potomac Drive
King George, VA 22485-5824

Phone: +1 540.644.6224
Fax: +1 540.663.0276
e-mail: pcroll@csc.com