SOFTWARE ASSURANCE FORUM

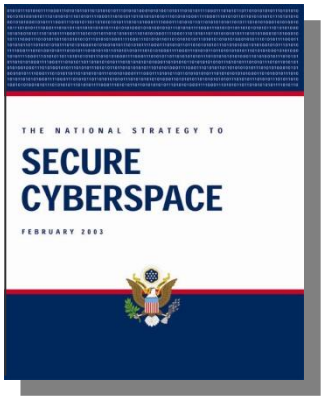
BUILDING SECURITY IN

Homeland Security

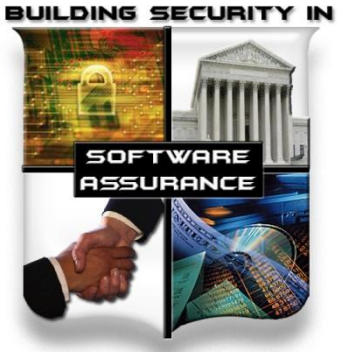
Commerce

National Defense

Next SwA Forum 27 Sep – 1 Oct 2010 at NIST, Gaithersburg, MD



Software Assurance



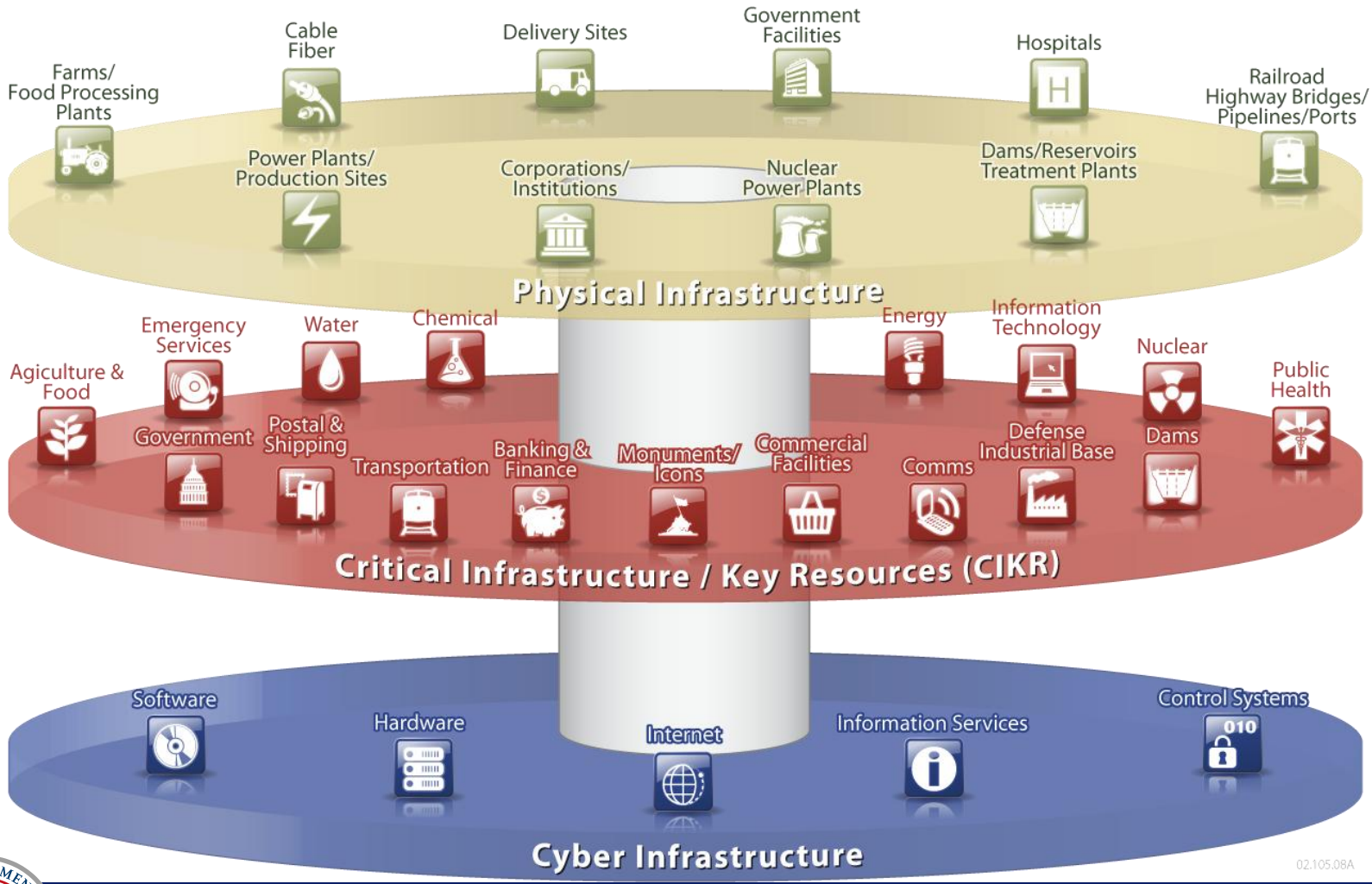
Public/Private Collaboration Efforts Mitigating Exploitable Software Risks throughout the Lifecycle

Sept 21, 2010

Joe Jarzombek, PMP, CSSLP
Director for Software Assurance
National Cyber Security Division
Office of the Assistant Secretary for
Cybersecurity and Communications

Homeland Security

Interdependencies Between Physical and Cyber Infrastructures -- Need for secure software applications

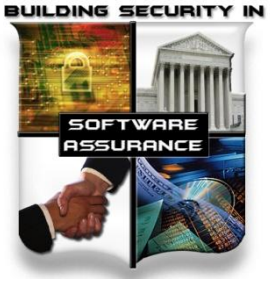


DHS NCSD Software Assurance (SwA) Program

Through public-private collaboration promotes security and resilience of software throughout the lifecycle; focused on reducing exploitable software weaknesses and addressing means to improve capabilities that routinely develop, acquire, and deploy resilient software products. Collaboratively advancing software-relevant rating schemes

- **Serves as a focal point for interagency public-private collaboration to enhance development and acquisition processes and capability benchmarking to address software security needs.**
  - Hosts interagency Software Assurance Forums, Working Groups and training to provide public-private collaboration in advancing software security and providing publicly available resources.
  - Provides collaboratively developed, peer-reviewed information resources on Software Assurance, via journals, guides & on-line resources suitable for use in education, training, and process improvement.
  - Provides input and criteria for leveraging international standards and maturity models used for process improvement and capability benchmarking of software suppliers and acquisition organizations.
- **Enables software security automation and measurement capabilities through use of common indexing and reporting capabilities for malware, exploitable software weaknesses, and common attacks which target software.**
  - Collaborates with the National Institute of Standards and Technology, international standards organizations, and tool vendors to create standards, metrics and certification mechanisms from which tools can be qualified for software security verification.
  - Manages programs to facilitate the adoption of Malware Attribute Enumeration Classification (MAEC), Common Weakness Enumeration (CWE), and Common Attack Pattern Enumeration and Classification (CAPEC).

Critical Considerations

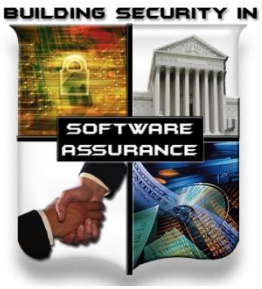


- Software is the core constituent of modern products and services – it enables functionality and business operations
- Dramatic increase in mission risk due to increasing:
  - Software dependence and system interdependence (weakest link syndrome)
  - Software Size & Complexity (obscures intent and precludes exhaustive test)
  - Outsourcing and use of un-vetted software supply chain (COTS & custom)
  - Attack sophistication (easing exploitation)
  - Reuse (unintended consequences increasing number of vulnerable targets)
  - Number of vulnerabilities & incidents with threats targeting software
  - Risk of Asymmetric Attack and Threats
- Increasing awareness and concern

Homeland Security

Software and the processes for acquiring and developing software represent a material weakness

Security-Enhanced Capabilities:
Mitigating Risks to the Enterprise



- With today's global software supply chain, Software Engineering, Quality Assurance, Testing and Project Management must explicitly address security risks posed by exploitable software.
  - Traditional processes do not explicitly address software-related security risks that can be passed from projects to using organizations.
- Mitigating Supply Chain Risks requires an understanding and management of Suppliers' Capabilities, Products and Services
  - Enterprise risks stemming from supply chain are influenced by suppliers and acquisition projects (including procurement, SwEng, QA, & testing).
  - IT/Software Assurance processes/practices span development/acquisition.
  - Derived (non-explicit) security requirements should be elicited/considered.
- More comprehensive diagnostic capabilities and standards are needed to support processes and provide transparency for more informed decision-making for mitigating risks to the enterprise

Homeland Security

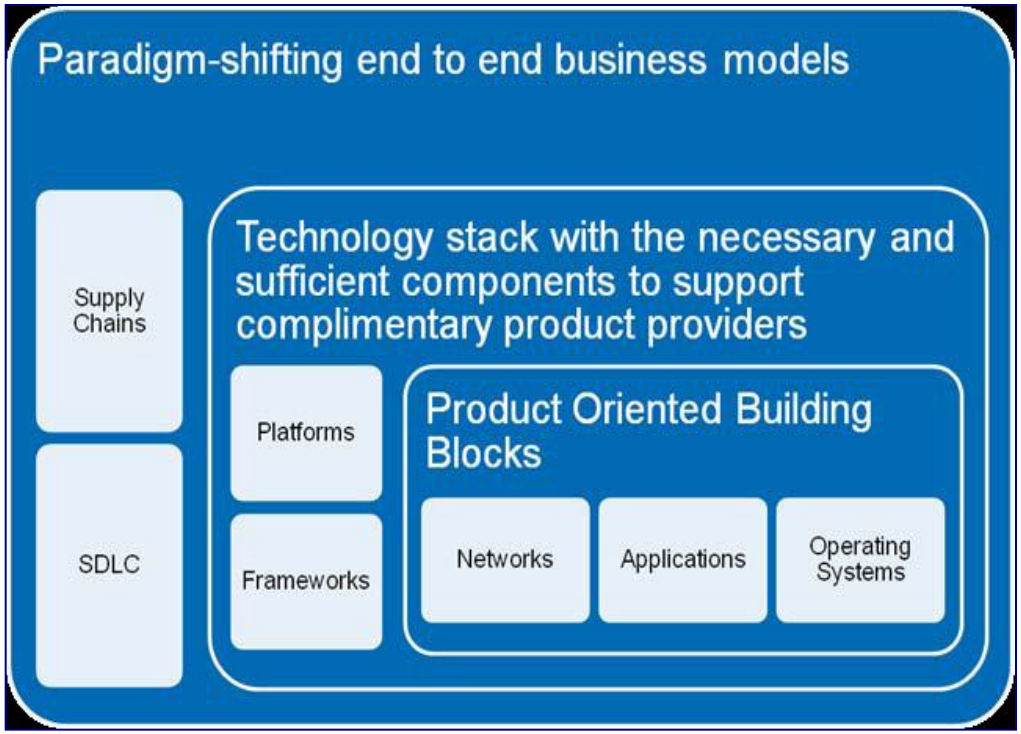
Free resources are available to assist personnel in security-enhancing contracting, outsourcing and development activities (see https://buildsecurityin.us-cert.gov)

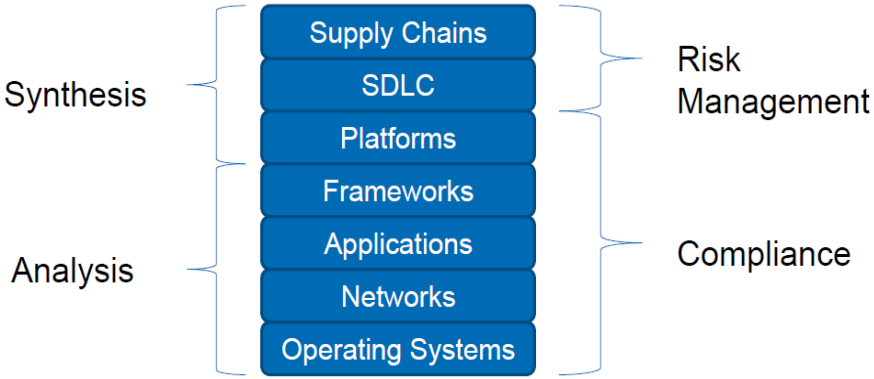
IT/software security risk landscape is a convergence between “defense in depth” and “defense in breadth”

Enterprise Risk Management and Governance are security motivators

Acquisition could be considered the beginning of the lifecycle; not development

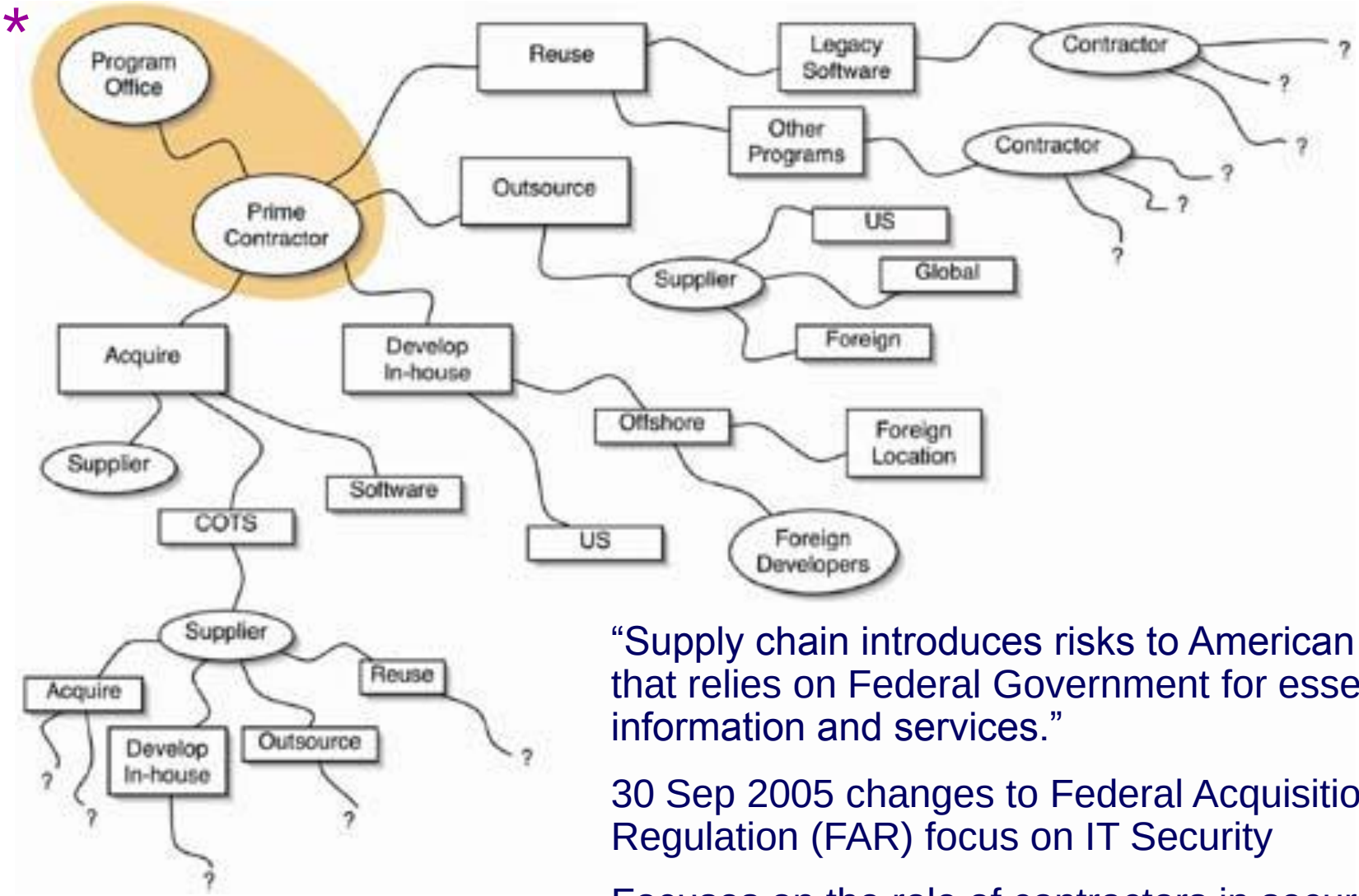
> “In the digital age, sovereignty is demarcated not by territorial frontiers but by supply chains.”
>
> – Dan Geer, CISO In-Q-Tel





Software Assurance provides a focus for:
-- Secure Software Components,
-- Security in the Software Life Cycle and
-- Software Supply Chain Risk Management

*



"Supply chain introduces risks to American society that relies on Federal Government for essential information and services."

30 Sep 2005 changes to Federal Acquisition Regulation (FAR) focus on IT Security

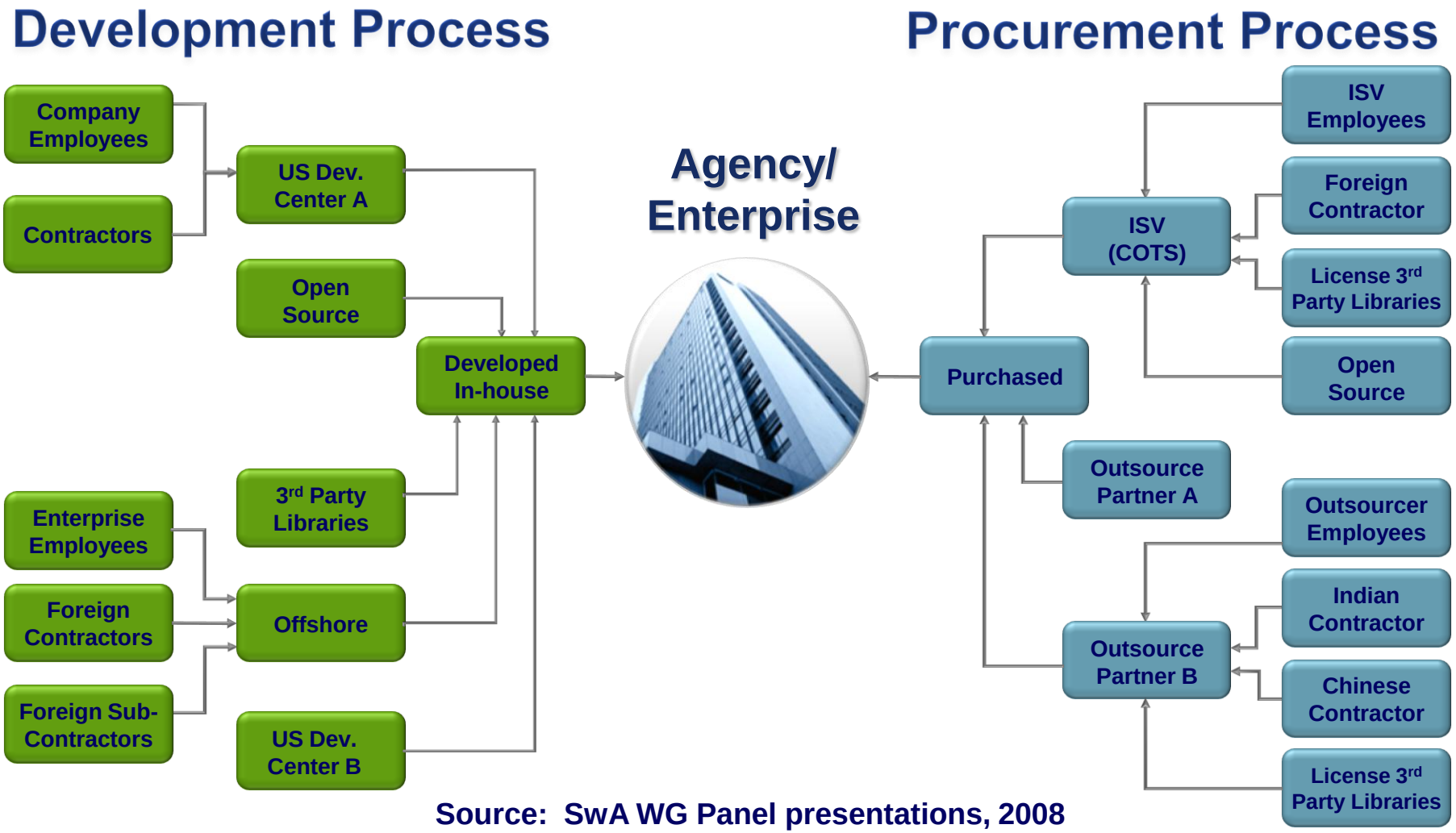
Focuses on the role of contractors in security as Federal agencies outsource various IT functions.

Homeland Security

"Scope of Supplier Expansion and Foreign Involvement" graphic in DACS www.softwaretechnews.com Secure Software Engineering, July 2005 article "Software Development Security: A Risk Management Perspective" synopsis of May 2004 GAO-04-678 report "Defense Acquisition: Knowledge of Software Suppliers Needed to Manage Risks"

Enterprise Processes for deploying capabilities: Increasingly Distributed and Complex

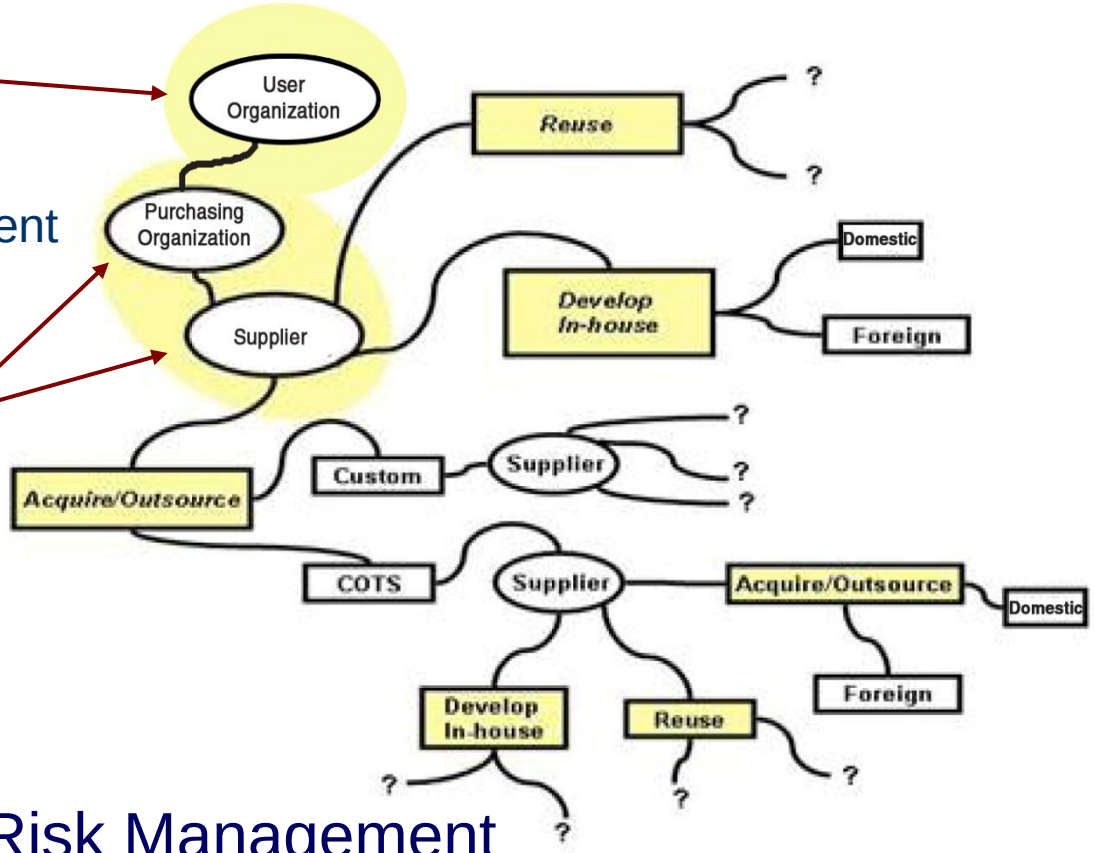
New Considerations for Quality & Security



Source: SwA WG Panel presentations, 2008

Risk Management (Enterprise <=> Project):
Shared Processes & Practices // Different Focuses

- Enterprise-Level:
  - Regulatory compliance
  - Changing threat environment
  - Business Case
- Program/Project-Level:
  - Cost
  - Schedule
  - Performance

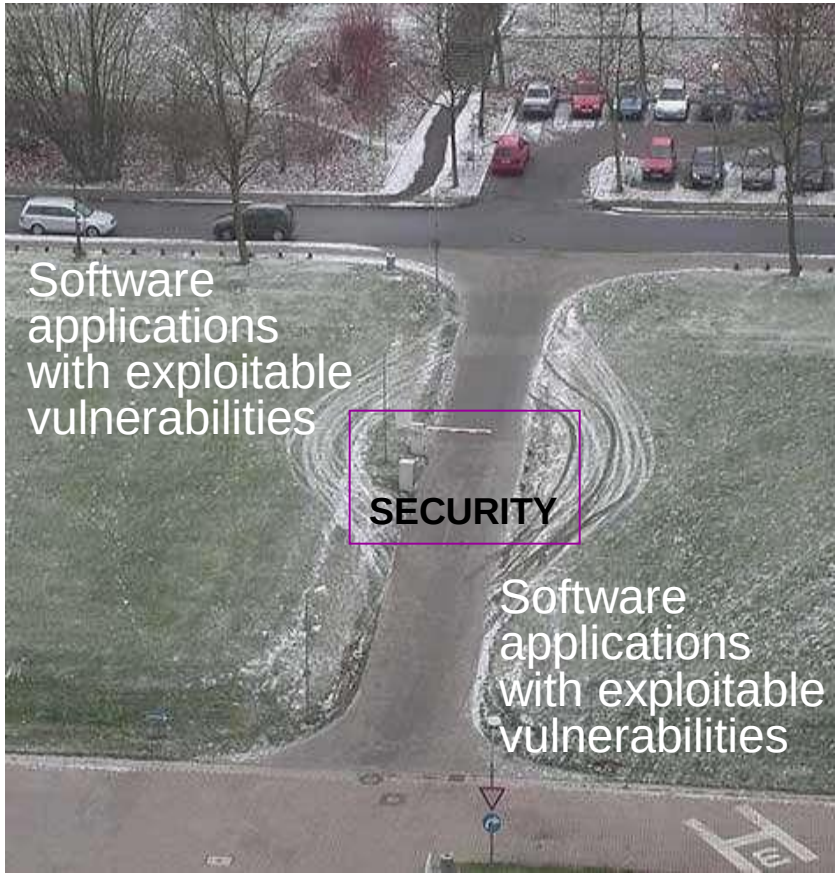


Software Supply Chain Risk Management traverses enterprise and program/project interests

Homeland Security

Security is a Requisite Quality Attribute:
Vulnerable Software Enables Exploitation

- Rather than attempt to break or defeat network or system security, hackers are opting to target application software to circumvent security controls.
  - **75% of hacks occurred at application level**
    - "90% of software attacks were aimed at application layer" (Gartner & Symantec, June 2006)
  - most exploitable software vulnerabilities are attributable to non-secure coding practices (and not identified in testing).
- Functional correctness must be exhibited even when software is subjected to abnormal and hostile conditions



In an era riddled with asymmetric cyber attacks, claims about system reliability, integrity & safety must include provisions for built-in security of the enabling software.

Homeland Security

Software Assurance "End State" Objectives…

- **Government, in collaboration with industry / academia, raised expectations for product assurance with requisite levels of integrity and security:**
  - Helped advance more comprehensive software assurance diagnostic capabilities to mitigate risks stemming from exploitable vulnerabilities and weaknesses;
  - Collaboratively advanced use of software security measurement & benchmarking schemes
  - Promoted use of methodologies and tools that enabled security to be part of normal business.
- **Acquisition managers & users factored risks posed by the software supply chain as part of the trade-space in risk mitigation efforts:**
  - Information on suppliers' process capabilities (business practices) would be used to determine security risks posed by the suppliers' products and services to the acquisition project and to the operations enabled by the software.
  - Information about evaluated products would be available, along with responsive provisions for discovering exploitable vulnerabilities, and products would be securely configured in use.
- **Suppliers delivered quality products with requisite integrity and made assurance claims about the IT/software safety, security and dependability:**
  - Relevant standards would be used from which to base business practices & make claims;
  - Qualified tools used in software lifecycle enabled developers/testers to mitigate security risks;
  - Standards and qualified tools would be used to certify software by independent third parties;
  - IT/software workforce had requisite knowledge/skills for developing secure, quality products.

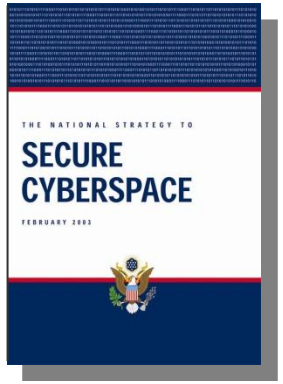
Need for Rating Schemes

- Rating of Software products:
  - Supported by automation
  - Standards-based
  - Rules for aggregation and scaling
  - Verifiable by independent third parties
  - Labeling to support various needs (eg., security, dependability, etc)
  - Meaningful and economical for consumers and suppliers
- Rating of Suppliers providing software products and services
  - Standards-based or model-based frameworks to support process improvement and enable benchmarking of organizational capabilities
  - Credential programs for professionals involved in software lifecycle activities and decisions

DHS Software Assurance Program Overview

- Program established in response to the National Strategy to Secure Cyberspace - Action/Recommendation 2-14:

  "DHS will facilitate a national public-private effort to promulgate best practices and methodologies that promote integrity, security, and reliability in software code development, including processes and procedures that diminish the possibilities of erroneous code, malicious code, or trap doors that could be introduced during development."

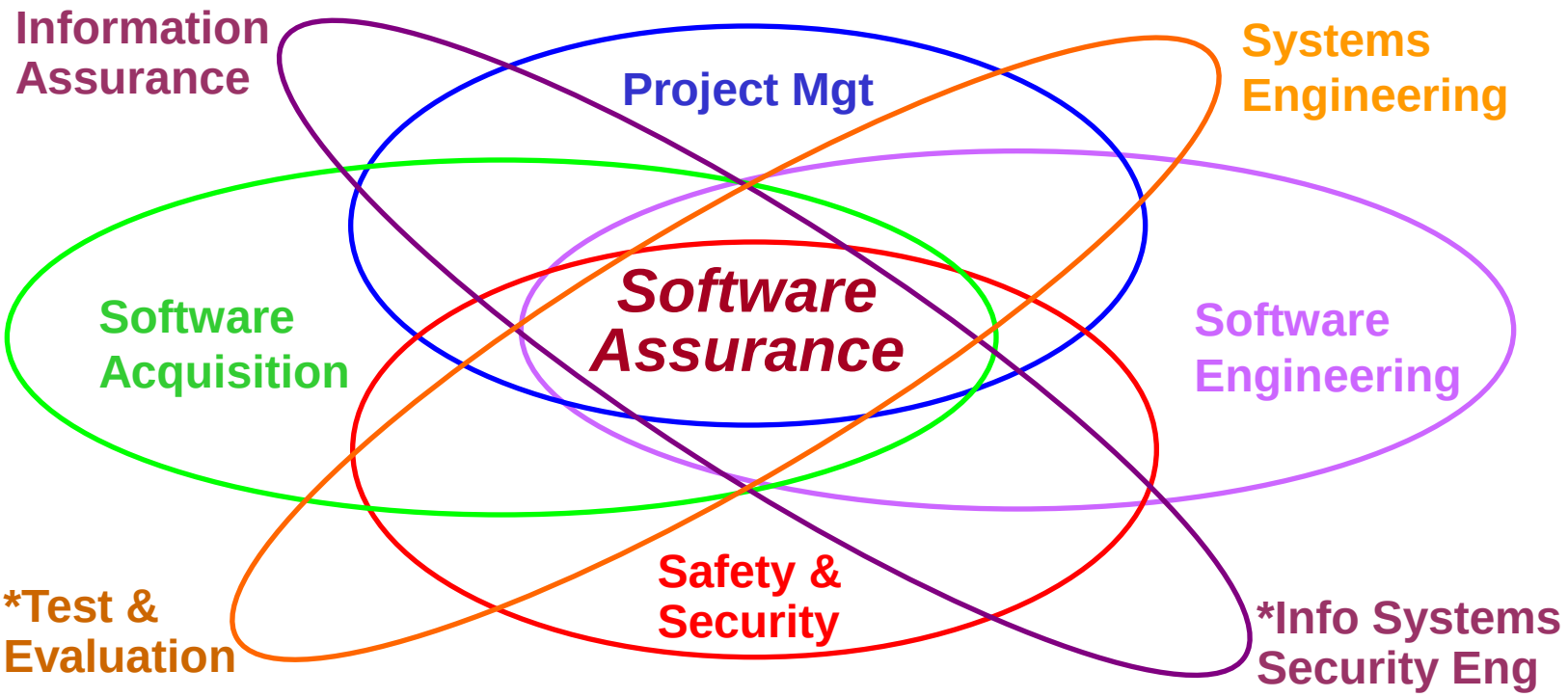


- DHS Program goals promote the **security and resilience** of software across the development, acquisition, and operational life cycle
- DHS Software Assurance (SwA) program is scoped to address:
  - **Trustworthiness** - No exploitable vulnerabilities or malicious logic exist in the software, either intentionally or unintentionally inserted,
  - **Dependability (Correct and Predictable Execution)** - Justifiable confidence that software, when executed, functions as intended,
  - **Survivability** - If compromised, damage to the software will be minimized; it will recover quickly to an acceptable level of operating capacity; it's 'rugged';
  - **Conformance** – Planned, systematic set of multi-disciplinary activities that ensure processes/products conform to requirements, standards/procedures.

Homeland Security

See Wikipedia.org for "Software Assurance" - CNSS Instruction No. 4009, "National Information Assurance Glossary," Revised 2006, defines Software Assurance as: "the level of confidence that software is free from vulnerabilities, either intentionally designed into the software or accidentally inserted at anytime during its lifecycle, and that the software functions in the intended manner".

Disciplines Contributing to Software Assurance*



In Education and Training, Software Assurance could be addressed as:

- A "knowledge area" extension within each of the contributing disciplines;
- A stand-alone CBK drawing upon contributing disciplines;
- A set of functional roles, drawing upon a common body of knowledge; allowing more in-depth coverage dependent upon the specific roles.

Intent is to provide framework for curriculum development and evolution of contributing BOKs

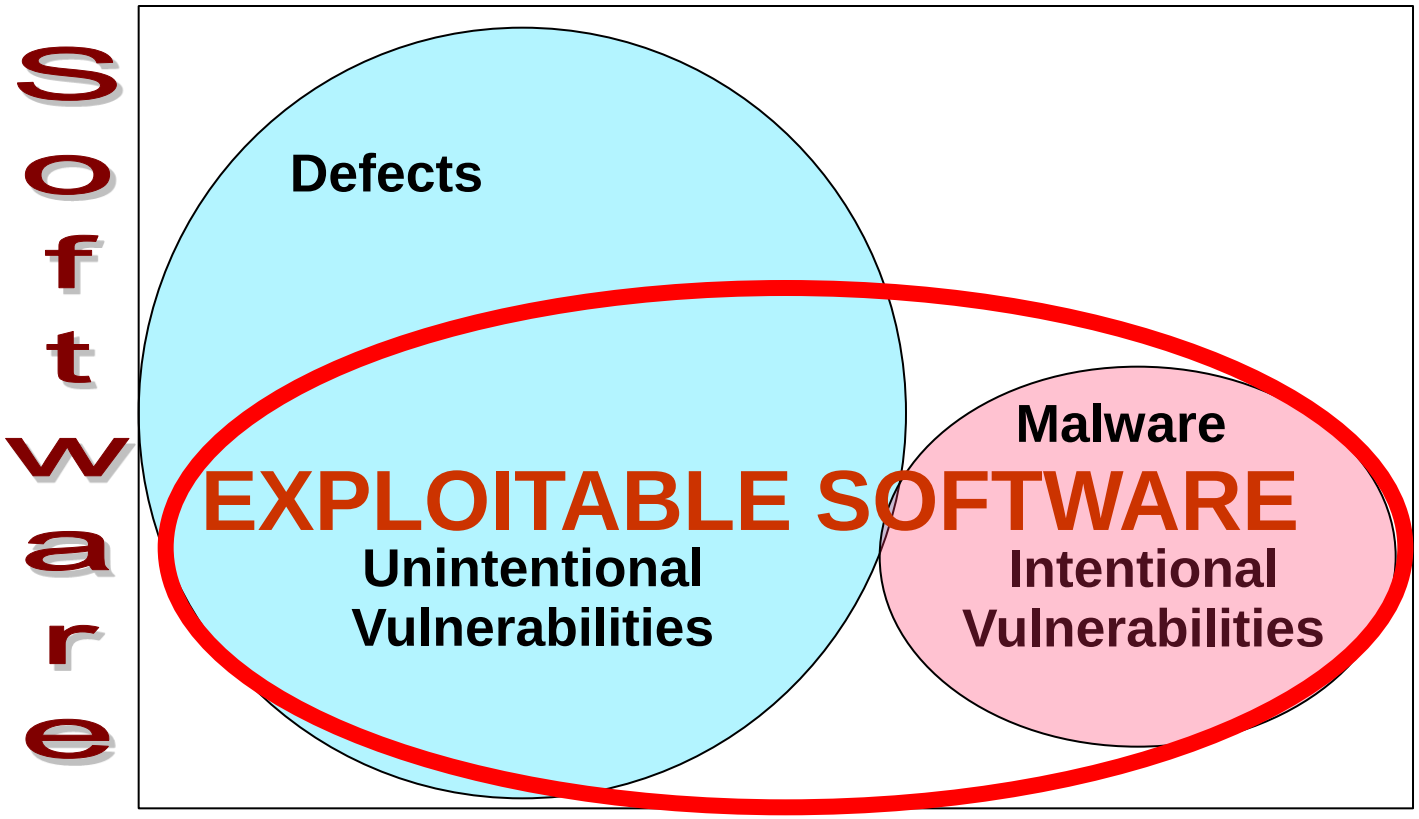
Homeland Security

* See 'Notes Page' view for contributing BOK URLs and relevant links

The intent is not to create a new profession of Software Assurance; rather, to provide a common body of knowledge: (1) from which to provide input for developing curriculum in related fields of study and (2) for evolving the contributing disciplines to better address the needs of software security, safety, dependability, reliability and integrity.

Software Assurance Addresses Exploitable Software:
Outcomes of non-secure practices and/or malicious intent

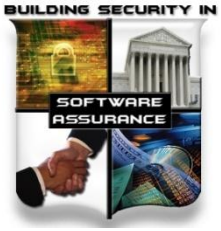
Exploitation potential of vulnerability is independent of "intent"



'High quality' can reduce security flaws attributable to defects; yet traditional S/W quality assurance does not address intentional malicious behavior in software

*Intentional vulnerabilities: spyware & malicious logic deliberately imbedded (might not be considered defects)

DHS Software Assurance Program Structure *

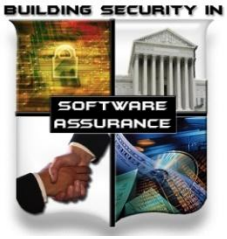


- As part of the DHS risk mitigation effort, the SwA Program seeks to reduce software vulnerabilities, minimize exploitation, and address ways to improve the routine development of trustworthy software products and tools to analyze systems for hidden vulnerabilities.
- The SwA framework encourages the production, evaluation and acquisition of better quality and more secure software; leverages resources to target the following four areas:
  - **People** – education and training for developers and users
  - **Processes** – sound practices, standards, and practical guidelines for the development of secure software
  - **Technology** – diagnostic tools, cyber security R&D and measurement
  - **Acquisition** – due-diligence questionnaires, contract templates and guidelines for acquisition management and outsourcing

Homeland Security

* July 28, 2006 statement of George Foresman, DHS UnderSecretary for Preparedness, before the U.S. Senate Committee on Homeland Security and Governmental Affairs, Subcommittee on Federal Financial Management, Government Information, and International Security

Software Assurance Forum & Working Groups*



… encourage the production, evaluation and acquisition of better quality and more secure software through targeting

People	Processes	Technology	Acquisition
Developers and users education & training	Sound practices, standards, & practical guidelines for secure software development	Security test criteria, diagnostic tools, common enumerations, SwA R&D, and SwA measurement	Software security improvements through due-diligence questions, specs and guidelines for acquisitions/ outsourcing

Products and Contributions

Build Security In - https://buildsecurityin.us-cert.gov and SwA community resources & info clearinghouse

SwA Common Body of Knowledge (CBK) & Glossary
Organization of SwSys Security Principles/Guidelines
SwA Developers' Guide on Security-Enhancing SDLC

Software Security Assurance State of the Art Report
Systems Assurance Guide (via DoD and NDIA)

SwA-related standards – ISO/IEC JTC1 SC7/27/22, IEEE CS, OMG, TOG, & CMM-based Assurance

Practical Measurement Framework for SwA/InfoSec
Making the Business Case for Software Assurance

SwA Metrics & Tool Evaluation (with NIST)
SwA Ecosystem w/ DoD, NSA, NIST, OMG & TOG
NIST Special Pub 500 Series on SwA Tools

Common Weakness Enumeration (CWE) dictionary
Common Attack Pattern Enumeration (CAPEC)

SwA in Acquisition: Mitigating Risks to Enterprise
Software Project Management for SwA SOAR

Homeland Security

* SwA Forum is part of Cross-Sector Cyber Security Working Group (CSCSWG) established under auspices of the Critical Infrastructure Partnership Advisory Council (CIPAC) that provides legal framework for participation.

DHS Software Assurance (SwA) Outreach

- Co-sponsor quarterly SwA WG sessions and semi-annual Software Assurance Forum for government, academia, and industry to facilitate ongoing public-private collaboration
- Co-sponsor SwA issues of CROSSTALK to "spread the word"
  - March 2007 issue on "Software Security"
  - May 2007 issue on "Software Acquisition"
  - Sep 2007 issue on "Service Oriented Architecture"
  - June 2008 issue on "Software Quality"
  - Sep 2008 issue on "Application Security"
  - Mar/Apr 2009 issue on "Reinforcing Good Practices"
  - Sep/Oct 2009 issue on "Resilient Software"
  - Mar/Apr 2010 issue on "Systems Assurance"
  - Sep/Oct 2010 issue on "Game Changing Tools & Practices"
- Provide outreach via DHS Speakers Bureau
- Collaborate with standards organizations, consortiums and professional societies in promoting SwA and participate in on-line communities, such as LinkedIn SwA mega-community
- Provide free SwA resources via "BuildSecurityIn" website to promote secure development methodologies (since Oct 05)
- Host Software Assurance Community Resources & Information Clearinghouse for SwA mega-community via https://buildsecurityin.us-cert.gov/SwA (since Dec 07)

Homeland Security

SwA Collaboration for Content & Peer Review

BSI https://buildsecurityin.us-cert.gov focuses on making Software Security a normal part of Software Engineering

SwA Community Resources and Information Clearinghouse (CRIC)

https://buildsecurityin.us-cert.gov/swa/ focuses on all contributing disciplines, practices and methodologies that advance risk mitigation efforts to enable greater resilience of software/cyber assets.

The SwA CRIC provides a primary resource for SwA Working Groups.

Where applicable, SwA CRIC & BSI provide relevant links to each other.

Software Assurance

Community Resources and Information Clearinghouse

Sponsored by DHS National Cyber Security Division

SwA Working Groups
- Workforce Education & Training
- Processes & Practices
- Technology, Tools & Product Eval.
- Acquisition & Outsourcing
- Measurement
- Business Case
- Malware

SwA Communities

SwA Forums

SwA Landscape

US-CERT Software Assurance

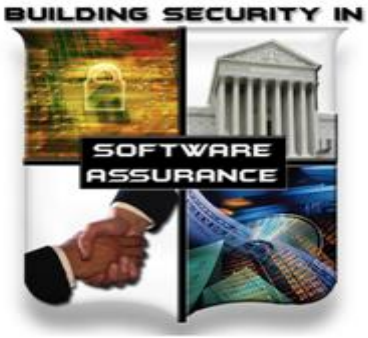
Build Security In

Homeland Security

Software assurance (SwA) is the level of confidence that software is free from vulnerabilities, either intentionally designed into the software or accidentally inserted at any time during its life cycle, and that the software functions in the intended manner (from CNSS 4009 IA Glossary - see Wikipedia for definitions and descriptions).

As part of DHS risk mitigation efforts to enable greater resilience of cyber assets, the Software Assurance Program seeks to reduce software vulnerabilities, minimize exploitation, and address ways to routinely acquire, develop and deploy reliable and trustworthy software products with predictable execution, and to improve diagnostic capabilities to analyze systems for exploitable weaknesses.

The **Software Assurance Forum** and several **working groups**, composed of stakeholders in government, industry, and academia, are contributing to efforts focused on advancing software assurance objectives. Anyone can participate in these public/private collaboration activities. Information about upcoming SwA Forums and Working Group sessions is posted on the SwA Forums page as it becomes available.



Focused efforts for advancing software assurance are addressed in the working groups listed below. Click on any working group's name to see **Recent Releases and Updates**, current activities, and other information for that working group.

- Workforce Education & Training
- Processes & Practices
- Technology, Tools & Product Evaluation
- Acquisition & Outsourcing
- Measurement
- Business Case
- Malware

WHY IS SOFTWARE ASSURANCE CRITICAL?

The nation's critical infrastructure (energy, transportation, telecommunications, etc.), businesses, and services are extensively and increasingly controlled and enabled by software. Vulnerabilities in that software put those resources at risk. The risk is

Life-Cycle Standards View Categories (ISO/IEC 15288 and 12207)

Organization

Governance Processes

Strategy and policy

Enterprise risk management
- Compliance
- Business case

Supply Chain Management

Project-Enabling Processes

Life Cycle Model Management

Infrastructure Management
- SwA ecosystem
- Enumerations, languages, and repositories

Project Portfolio Management

Human Resource Management
- SwA education
- SwA certification and training
- Recruitment

Quality Management

Agreement Processes

Acquisition
- Outsourcing
- Agreements
- Risk-based due diligence
- Supplier assessment

Supply

Project

Project Management Processes

Project Planning

Project Assessment and Control
- Assurance case management

Project Support Processes

Decision Management

Risk Management
- Threat Assessment

Configuration Management

Information Management

Measurement

Engineering

Technical Processes

Stakeholder Requirements Definition

Requirements Analysis
•Attack modeling (misuse and abuse cases)
•Data and information classification
•Risk-based derived requirements
•Sw security requirements

Architectural Design
•Secure Sw architectural design
•Risk-based architectural analysis
•Secure Sw detailed design and analysis

Implementation
•Secure coding and Sw construction
•Security code review and static analysis
•Formal methods

Integration
•Sw component integration
•Risk analysis of Sw reuse components

Verification & Validation
•Risk-based test planning
•Security-enhanced test and evaluation
- Dynamic and static code analysis
- Penetration testing

•Independent test and certification

Transition
•Secure distribution and delivery
•Secure software environment (secure configuration, application monitoring, code signing, etc)

Operations and Sustainment

Operation
- Incident handling and response

Maintenance
- Defect tracking and remediation
- Vulnerability and patch management
- Version control and management

Disposal

Software Reuse Processes

Domain Engineering

Reuse Asset Management

Reuse Program Management

Software Support Processes

Sw Documentation Management

Sw Quality Assurance

Sw Configuration Management

Sw Verification & Sw Validation

Sw Review

Sw Audit

Sw Problem Resolution


Build Security In Home - Windows Internet Explorer
https://buildsecurityin.us-cert.gov/bsi/home.html
Live Search
File Edit View Favorites Tools Help
Build Security In Home
Page Tools
Homeland Security
Privacy and Use
Build Security In
Setting a higher standard for software assurance
Sponsored by DHS National Cyber Security Division
Actions
Search BSI:
Search
Navigational Links
Home
Mission
Articles [by Content Area]
Secure Coding Sites
DHS SwA Web Site
DHS Software Assurance Resources
Additional Resources
Events
About Us
FAQs
RSS Feeds
Contact Us
Build Security In Home
What is Build Security In?
Build Security In is a collaborative effort that provides practices, tools, guidelines, rules, principles, and other resources that software developers, architects, and security practitioners can use to build security into software in every phase of its development.
Introduction to Software Security
Call for Authors and Reviewers
Submit an article for publication on BSI or volunteer to review new articles. See the Call for Authors and Reviewers for details.
Community Collaboration
To access other software assurance materials or to join the collaboration efforts of a related working group, visit the DHS Software Assurance Program's Community Resources and Information Clearinghouse.
Sponsor and Contributors
Build Security In is a Software Assurance strategic initiative of the National Cyber Security Division (NCSD) of the U.S. Department of Homeland Security. Peer-
Improve Security and Software Assurance: Tackle the CWE Top 25 – The Most Dangerous Programming Errors
The Top 25 CWEs represent the most significant exploitable software constructs that have made software so vulnerable. Addressing these will go a long way in securing software, both in development and in operation. Read more and see the list of Top 25 CWE Programming Errors on the Software Assurance Community Resources and Information Clearinghouse website.
Consistent with this list is the Top 10 for 2010 released by the Open Web Application Security Project (OWASP). OWASP's report captures the top ten risks associated with the use of web applications in an enterprise. Download the report, which contains examples and details that explain these risks to software developers, managers, and anyone interested in the future of web security, for free here.
What's New
Calls for papers have been posted for the Second IEEE International Conference on Information Privacy, Security, Risk, and Trust (PASSAT), the 1st International Workshop on Measurability of Security in Software Architectures (MeSSa), and the Software Assurance Minitrack of the 44th Hawaii International Conference on System Sciences (HICSS-44).
A new article, Improving Software Assurance, has been added.
A new article, Supply-Chain Risk Management: Incorporating

Process Agnostic Lifecycle

Architecture & Design
- Architectural risk analysis
- Threat modeling
- Principles
- Guidelines
- Historical risks
- Modeling tools
- Resources

Code
- Code analysis
- Assembly, integration & evolution
- Coding practices
- Coding rules
- Code analysis
- Resources

Test
- Security testing
- White box testing
- Attack patterns
- Historical risks
- Resources

Requirements
- Requirements engineering
- Attack patterns
- Resources

Touch Points & Artifacts

Fundamentals
- Risk management
- Project management
- Training & awareness
- Measurement
- SDLC process
- Business relevance
- Resources

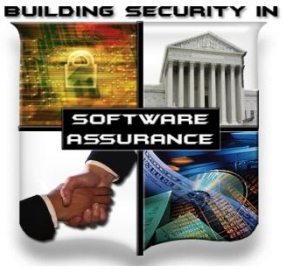
System
- Penetration testing
- Incident management
- Deployment & operations
- Black box testing
- Resources

Key
- Best (sound) practices
- Foundational knowledge
- Tools
- Resources

https://buildsecurityin.us-cert.gov

Homeland Security

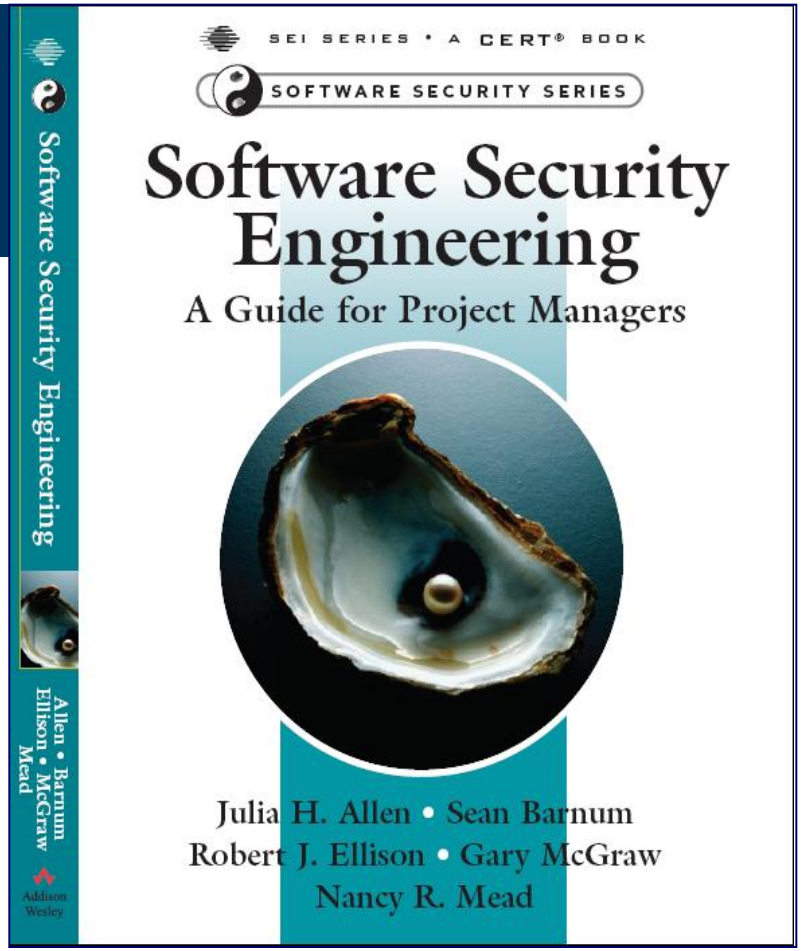
Software Security Engineering: A Guide for Project Managers



Build Security In

Setting a Higher Standard for Software Assurance

Sponsored by DHS National Cyber Security Division



- Organized for Project Managers
  - Derives material from DHS SwA "Build Security In" web site
    - https://buildsecurityin.us-cert.gov
  - Provides a process focus for projects delivering software-intensive products and systems
- Published in May 2008

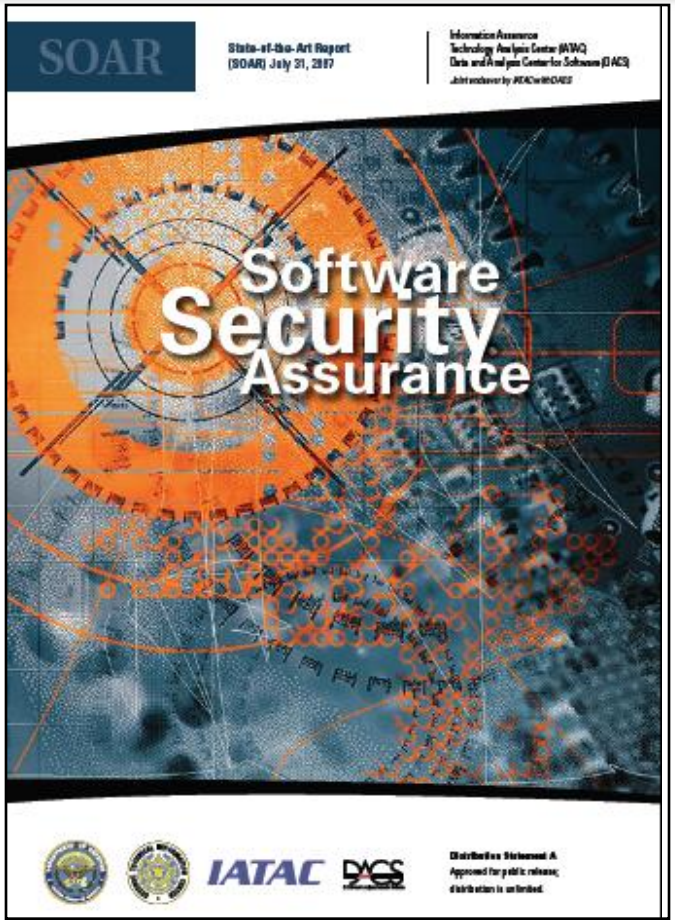
Homeland Security

SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

State of the Art Report

- July 2007 FREE publicly available resource provides a comprehensive look at efforts to improve the state of Software Security Assurance:
  - describes the threats and common vulnerabilities to which software is subject;
  - presents the many ways in which the S/W Security Assurance problem is being framed and understood across government, industry, and academia;
  - describes numerous methodologies, best practices, technologies, and tools currently being used to specify, design, and implement software that will be less vulnerable to attack, and to verify its attack-resistance, attack-tolerance, and attack-resilience;
  - offers a large number of available resources from which to learn more about principles and practices that constitute Software Security Assurance;
  - provides observations about potentials for success, remaining shortcomings, and emerging trends across the S/W Security Assurance landscape.
- Free via http://iac.dtic.mil/iatac/download/security.pdf



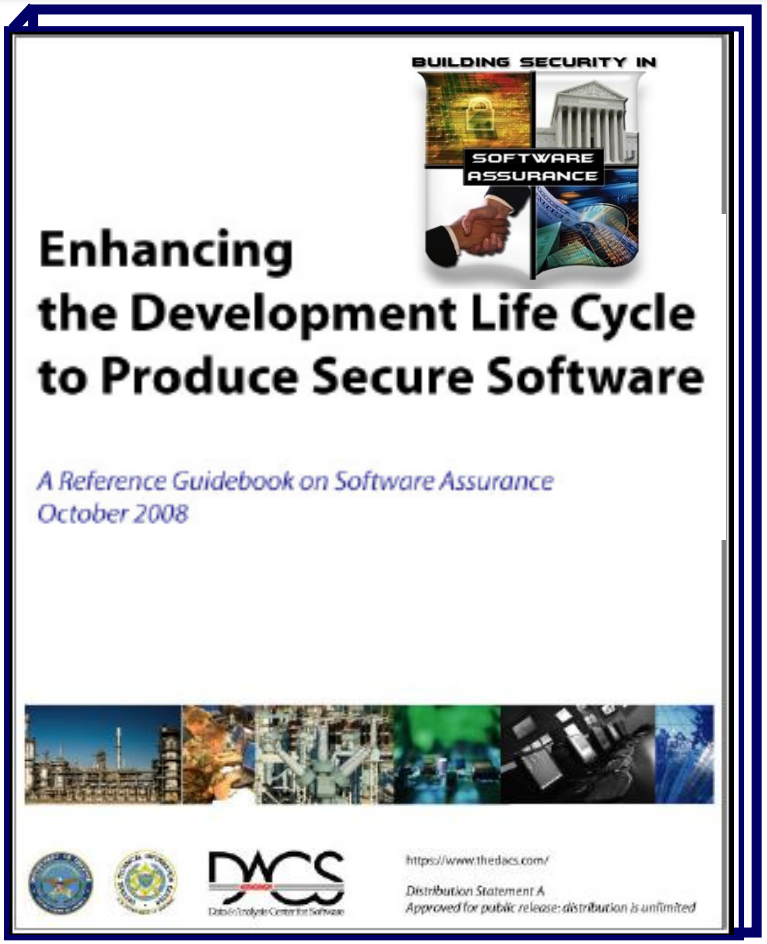
•The SOAR reflects output of efforts in the DoD-DHS Software Assurance Forum and Working Groups that provide collaborative venues for stakeholders to share and advance techniques and technologies relevant to software security.

SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

Reference Resource on Software Assurance

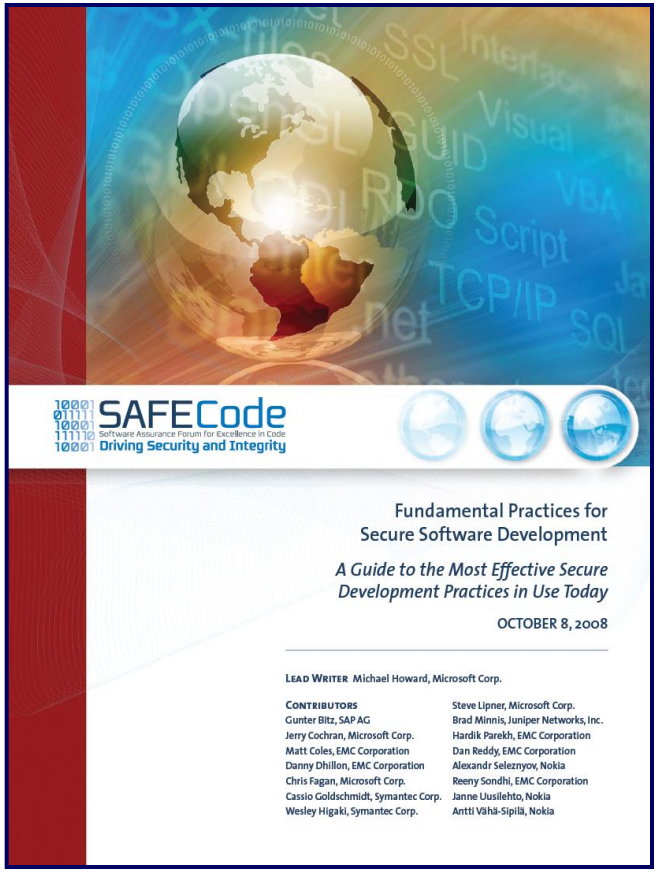
- Describes how to integrate security principles and practices in software development life cycle
- Addresses security requirements, secure design principles, secure coding, risk-based software security testing, and secure sustainment
- Provides guidance for selecting secure development methodologies, practices, and technologies
  - Collaboratively developed/updated via SwA Forum working groups
  - Released Oct 2008 by DACS
  - Free, available for download via DACS & DHS SwA Community Resources & Information Clearinghouse



https://www.thedacs.com/techs/enhanced_life_cycles/

Fundamental Practices for Secure Software Development:
A Guide to the Most Effective Secure Development Practices in Use Today, Oct 8, 2008

- Common security-related elements of software development methodologies
  - Security requirements help drive design, code handling, programming, and testing activities
- Secure Programming practices:
  - Minimize unsafe function use
  - Use the latest compiler toolset
  - Use static and dynamic analysis tools
  - Use manual code review on high-risk code
  - Validate input and output
  - Use anti-cross site scripting libraries
  - Use canonical data formats
  - Avoid string concatenation for dynamic SQL
  - Eliminate weak cryptography
  - Use logging and tracing
- Test to validate robustness and security
  - Fuzz testing
  - Penetration testing & third party assessment
  - Automated test tools (in all development stages)
- Code Integrity and Handling
  - Least privilege access, Separation of duties,
  - Persistent protection, Compliance management; Chain of custody & supply chain integrity.
- Documentation (about software security posture & secure configurations)



http://www.safecode.org/publications/SAFECode_Dev_Practices1008.pdf

SOFTWARE ASSURANCE FORUM

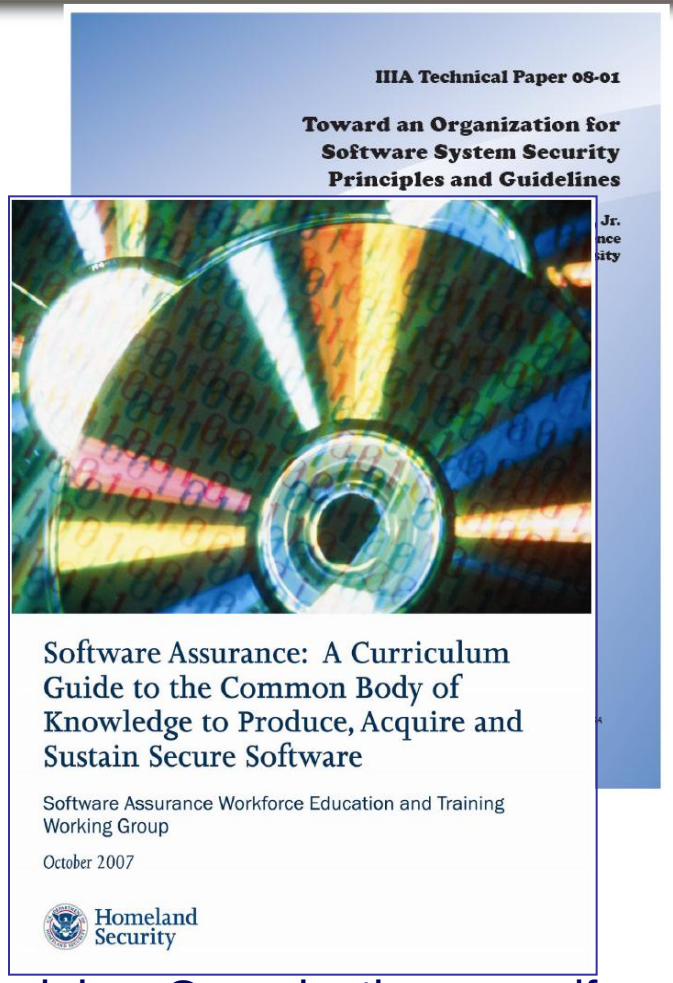
BUILDING SECURITY IN

Content for Curricula Development

- ***"Software Assurance: A Curriculum Guide to the Common Body of Knowledge to Produce, Acquire, and Sustain Secure Software,"*** updated Oct 2007

- ***"Toward an Organization for Software System Security Principles and Guidelines,"*** Version 1.0, IIIA Technical Paper 08-01. Feb 2008

Both collaboratively developed through the Software Assurance Working Group on Workforce Education and Training



http://www.jmu.edu/iiia/webdocs/Reports/SwA_Principles_Organization-sm.pdf

Software Assurance (SwA) Pocket Guide Series

SwA in Acquisition & Outsourcing

- Software Assurance in Acquisition and Contract Language
- Software Supply Chain Risk Management and Due-Diligence

SwA in Development

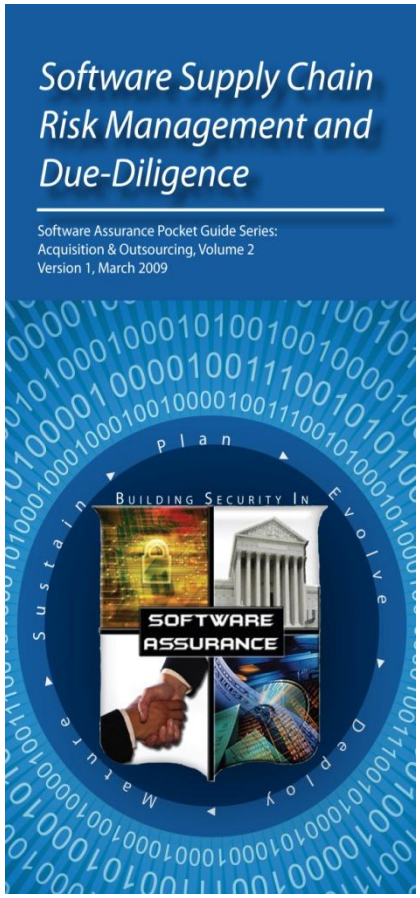
- Integrating Security into the Software Development Life Cycle
- Key Practices for Mitigating the Most Egregious Exploitable Software Weaknesses
- Risk-based Software Security Testing
- Requirements and Analysis for Secure Software
- Architecture and Design Considerations for Secure Software
- Secure Coding and Software Construction
- Security Considerations for Technologies, Methodologies & Languages

SwA Life Cycle Support

- SwA in Education, Training and Certification
- Secure Software Distribution, Deployment, and Operations
- Code Transparency & Software Labels
- Assurance Case Management
- Secure Software Environment and Assurance EcoSystem

SwA Measurement and Information Needs

- Making Software Security Measurable
- Practical Measurement Framework for SwA and InfoSec
- SwA Business Case and Return on Investment



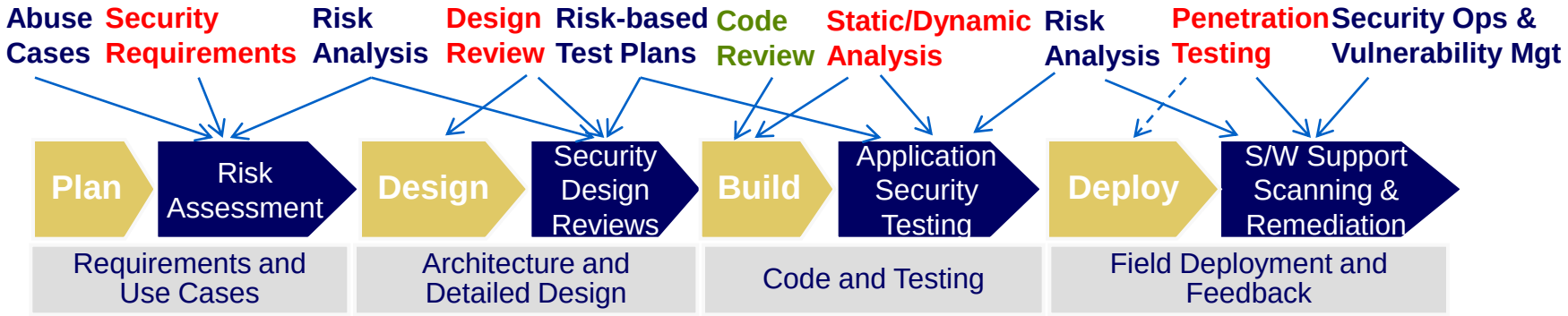
SwA Pocket Guides and SwA-related documents are collaboratively developed with peer review; they are subject to update and are freely available for download via the DHS Software Assurance Community Resources and Information Clearinghouse at https://buildsecurityin.us-cert.gov/swa (see SwA Resources)

Security-Enhanced Process Improvements

Organizations that provide security engineering & risk-based analysis throughout the lifecycle will have more resilient software products / systems.

"Build Security In" throughout the lifecycle

Attack Modeling	Secure S/W Requirements Engineering	Secure Design Principles & Practices	Secure Programming Practices	Test / Validation of Security & Resilience	Secure Distribution/ Deployment	Documentation for Secure Use & Configuration



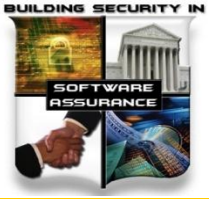
Organizational Process Assets cover: governance, policies, standards, training, tailoring guidelines

- Leverage Software Assurance resources (freely available) to incorporate in training & awareness
- Modify SDLC to incorporate security processes and tools (should be done in phases by practitioners to determine best integration points)
- Avoid drastic changes to existing development environment and allow for time to change culture and processes
- Make the business case and balance the benefits
- Retain upper management sponsorship and commitment to producing secure software.

Homeland Security

* Adopted in part from "Software Assurance: Mitigating Supply Chain Risks" (DHS NCSD SwA); "What to Test from a Security Perspective for the QA Professional" (Cigital) and "Neutralizing the Threat: A Case Study in Enterprise-wide Application Security Deployments" (Fortify Software & Accenture Security Technology Consulting)

Build Security In the SDLC



- Adding security practices throughout the SDLC establishes a software life cycle process that codifies both caution and intention.
- Key elements of a secure software life cycle process are:
  1. Security criteria in all software life cycle checkpoints (at entry & exit of a life cycle phase)
  2. Adherence to secure software principles and practices
  3. Adequate requirements, architecture, and design to address software security
  4. Secure coding practices with secure software integration/assembly practices
  5. Security testing practices that focus on verifying S/W dependability, trustworthiness, & resiliency
  6. Secure distribution and deployment practices and mechanisms
  7. Secure sustainment practices
  8. Supportive security tools (providing static & dynamic analysis) for developers and testers
  9. Secure software configuration management systems and processes
  10. Security risk analysis throughout the lifecycle
- Key people for producing secure software are:
  1. Security-knowledgeable software professionals
  2. Security-aware project management
  3. Upper management commitment to production of secure software

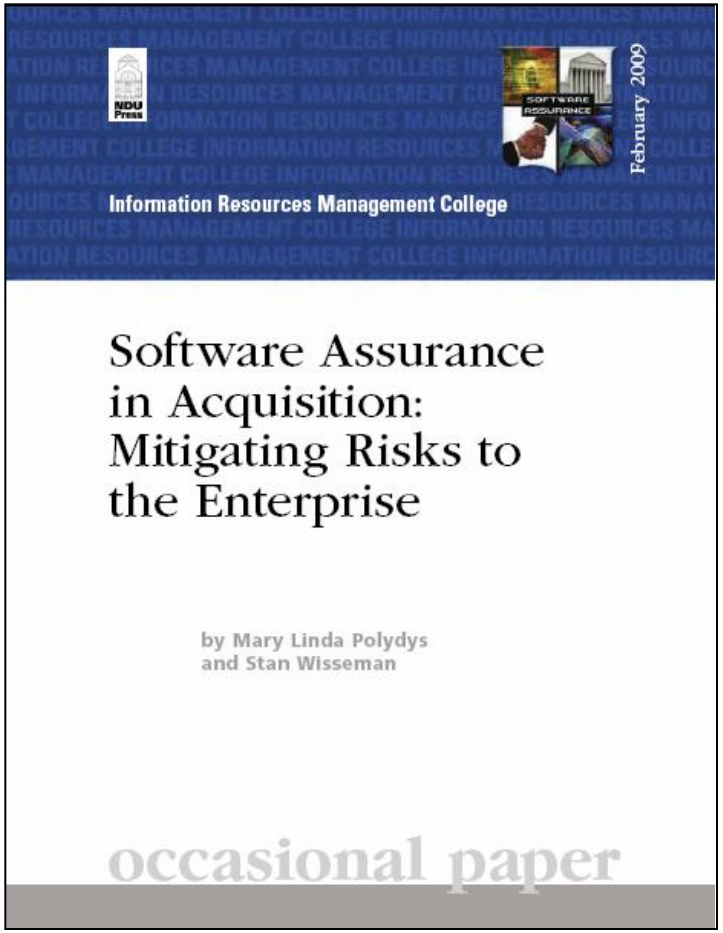
Homeland Security

Adopted from **Build Security In web site "Introduction to Software Security"** which adapted or excerpted from *Enhancing the Development Life Cycle to Produce Secure Software: A Reference Guidebook on Software Assurance* [DHS/DACS 08].

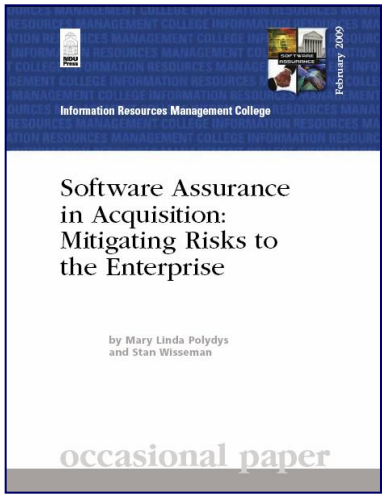
"Software Assurance in Acquisition:
Mitigating Risks to the Enterprise"

Version 1.0, Oct 2008, available for community use

published by National Defense University Press, Feb 2009



SwA Acquisition & Outsourcing Handbook



Software Supply Chain Risk Management and Due-Diligence -- Table 1 –SwA Concern Categories		
SwA Concern Categories	**Risks**	**Purpose for Questions**
Software History and Licensing Development Process Management Software Security Training and Awareness Planning and Requirements Architecture and Design Software Development Built-in Software Defenses Component Assembly Testing Software Manufacture and Packaging Installation Assurance Claims and Evidence Support Software Change Management Timeliness of Vulnerability Mitigation Individual Malicious Behavior Security "Track Record" Financial History and Status Organizational History Foreign Interests and Influences Service Confidentiality Policies Operating Environment for Services Security Services and Monitoring		

Software Supply Chain Risk Management and Due-Diligence -- *Table 1 –SwA Concern Categories*		
SwA Concern Categories	**Risks**	**Purpose for Questions**
Software History and Licensing	The software supplier's development practice in using code of unknown origin may be unable to produce trustworthy software.	To address supply chain concerns and identify risks pertaining to history/pedigree of software during any and all phases of its life cycle that should have been considered by the supplier.
Development Process Management	If supplier project management does not perceive the value of SwA and enforce best practices, they will not be consistently implemented.	To determine whether project management enforces software assurance–related best practices.
Software Security Training and Awareness	Developers unaware of software assurance best practices are likely to implement software with security flaws (making it more susceptible to attack).	To determine whether training of developers in SwA best practices is a supplier policy and practice.
Planning and Requirements	If nonfunctional requirements (security, quality, safety) are not specified, developers will not implement them.	To determine whether the supplier's requirements analysis process explicitly addresses SwA requirements.
Architecture and Design	The software may be designed without considering security or minimization of exploitable defects.	To determine how security is considered during the design phase.
Software Development	If developers lack qualified tools or if personnel are allowed to inappropriately access or change configuration items in the development environment, then delivered software might have unspecified features. The supplier might lack sufficient process capability to deliver secure products, systems or services.	To ascertain that the supplier has and enforces policies and SwA practices in the development of software that use secure software development environments to minimize risk exposures.
Built-in Software Defenses	The software may lack preventive measures to help it resist attack effectively and proactively.	To ensure that capabilities are designed to minimize the exposure of the software's vulnerabilities to external threats and to keep the software in a secure state regardless of the input and parameters it receives from its users or environment.

Software Supply Chain Risk Management and Due-Diligence -- Table 1 –SwA Concern Categories		
SwA Concern Categories	**Risks**	**Purpose for Questions**
Component Assembly	Insufficient analysis of software components used to assemble larger software packages may introduce vulnerabilities to the overall package.	To ensure that the software components are thoroughly vetted for their security properties, secure behaviors, and known types of weaknesses that can lead to exploitable vulnerabilities.
Testing	Software released with insufficient testing may contain an unacceptable number of exploitable defects.	To determine whether the appropriate set of analyses, reviews, and tests are performed on the software throughout the life cycle which evaluate security criteria.
Software Manufacture and Packaging	Vulnerabilities or malicious code could be introduced in the manufacturing or packaging process.	To determine how the software goes through the manufacturing process, how it is packaged, and how it remains secure.
Installation	The software may not install as advertised and the acquirer may not get the software to function as expected.	To ensure the supplier provides an acceptable level of support during the installation process.
Assurance Claims and Evidence	Supplier assurance claims (with supporting evidence) may be non-existent or insufficiently verified.	To determine how suppliers communicate their claims of assurance; ascertain what the claims have been measured against, and identify at what levels they will be verified.
Support	Supplier ceases to supply patches and new releases prior to the acquirer ending use of software. Vulnerabilities may go unmitigated.	To ensure understanding of supplier policy for security fixes and when products are no longer supported.
Software Change Management	Weak change control procedures can corrupt software and introduce new security vulnerabilities.	To determine whether software changes are adequately assessed and verified by supplier management.
Timeliness of Vulnerability Mitigation	Sometimes it can be extremely difficult to make a software supplier take notice and repair software to mitigate reported vulnerabilities.	To ensure security defects and configuration errors are fixed properly and in a timely fashion.

Software Supply Chain Risk Management and Due-Diligence -- *Table 1 –SwA Concern Categories*

SwA Concern Categories	Risks	Purpose for Questions
Individual Malicious Behavior	A developer purposely inserts malicious code, and supplier lacks procedures to mitigate risks from insider threats within the supply chain.	To determine whether the supplier has and enforces policies to minimize individual malicious behavior.
Security "Track Record"	A software supplier that is unresponsive to known software vulnerabilities may not mitigate/patch vulnerabilities in a timely manner.	To establish insight into whether the supplier places a high priority on security issues and will be responsive to vulnerabilities they will need to mitigate.
Financial History and Status	A software supplier that goes out of business will be unable to provide support or mitigate product defects and vulnerabilities.	To identify documented financial conditions or actions of the supplier that may impact its viability and stability, such as mergers, sell-offs, lawsuits, and financial losses.
Organizational History	There may be conflicting circumstances or competing interests within the organization that may lead to increased risk in the software development.	To understand the supplier's organizational background, roles, and relationships that might have an impact on supporting the software.
Foreign Interests and Influences	There may be controlling foreign interests (among organization officers or from countries) with malicious intent to the users' country or organization planning to use the software.	To help identify supplier companies that may have individuals with competing interests or malicious intent to a domestic buyer/user.
Service Confidentiality Policies	Without policies to enforce client data confidentiality/ privacy, acquirer's data could be at risk without service supplier liability.	To determine the service provider's confidentiality and privacy policies and ensure their enforcement.
Operating Environment for Services	Operating environment for the services may not be hardened or otherwise secure.	To understand the controls the supplier has established to operate the software securely.
Security Services and Monitoring	Insufficient security monitoring may allow attacks to impact services.	To ensure software and its operating environment are regularly reviewed for adherence to SwA requirements through periodic testing and evaluation.

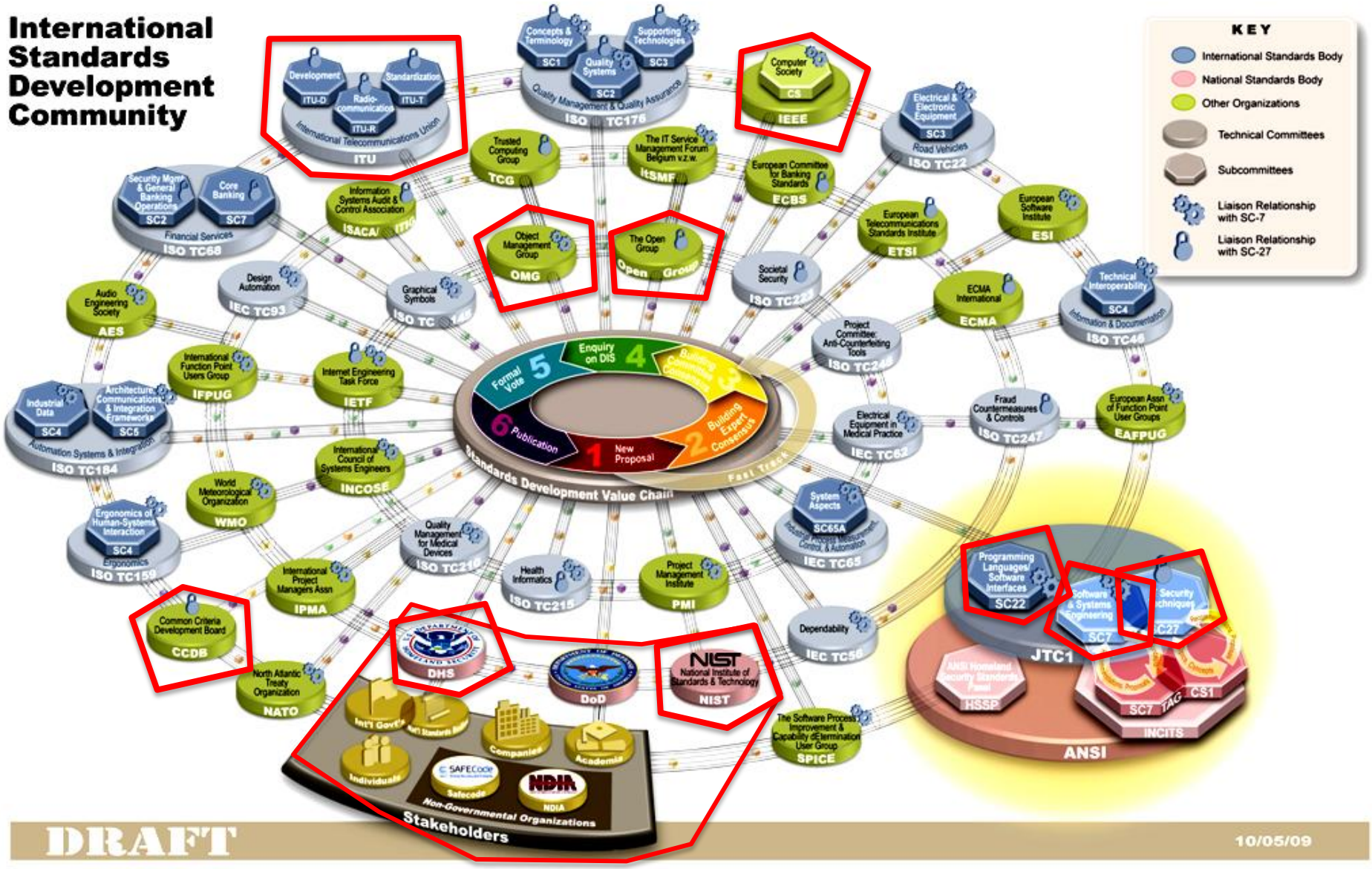
Table 2- Questions for COTS (Proprietary & Open Source), GOTS, & Custom Software					
No	Question	COTS Propri-etary	COTS Open-Source	GOTS	Custom
1	Can the pedigree of the software be established? Briefly explain what is known of the people and processes that created the software.	✓	✓	✓	✓
2	Explain the change management procedure that identifies the type and extent of changes conducted on the software throughout its life cycle.	✓		✓	✓
3	What type of license(s) are available for the open source software? Is it compatible with other software components in use? Is indemnification provided, and will the supplier indemnify the purchasing organization from any issues in the license agreement? Explain.	✓	✓		✓
4	Is there a clear chain of licensing from original author to latest modifier? Describe the chain of licensing.	✓			
5	What assurances are provided that the licensed software does not infringe upon any copyright or patent? Explain.	✓		✓	✓
6	Does the company have corporate policies and management controls in place to ensure that only corporate-approved (licensed and vetted) software components are used during the development process? Explain.	✓			✓
7	Are licensed software components still valid for the intended use?	✓		✓	
8	Is the software in question original source or a modified version?		✓		
9	Has the software been reviewed to confirm that it does not infringe upon any copyright or patent?	✓	✓		✓
10	How long has the software source been available? Is there an active user community providing peer review and actively evolving the software?	✓	✓		

Table 2- Questions for COTS (Proprietary & Open Source), GOTS, and Custom Software					
No.	Question	COTS Proprietary	COTS Open-Source	GOTS	Custom
11	Does the license/contract restrict the licensee from discovering flaws or disclosing details about software defects or weaknesses with others (e.g., is there a “gag rule” or limits on sharing information about discovered flaws)?	✓			✓
12	Does the license/contract restrict communications or limit the licensee in any potential communication with third-party advisors about provisions for support (e.g., is there a “gag rule” or limits placed on the licensee that affect ability to discuss contractual terms or breaches) regarding the licensed or contracted product or service?	✓			✓
13	Does software have a positive reputation? Does software have a positive reputation relative to security? Are there reviews that recommend it?	✓	✓		
14	Is the level of security where the software was developed the same as where the software will operate?			✓	✓
Development Process Management					
15	What are the processes (e.g., ISO 9000, CMMI, etc.), methods, tools (e.g., IDEs, compilers), techniques, etc. used to produce and transform the software (brief summary response)?	✓		✓	✓
16	What security measurement practices and data does the company use to assist product planning?	✓			✓
17	Is software assurance considered in all phases of development? Explain.	✓		✓	✓
18	How is software risk managed? Are anticipated threats identified, assessed, and prioritized?	✓		✓	✓

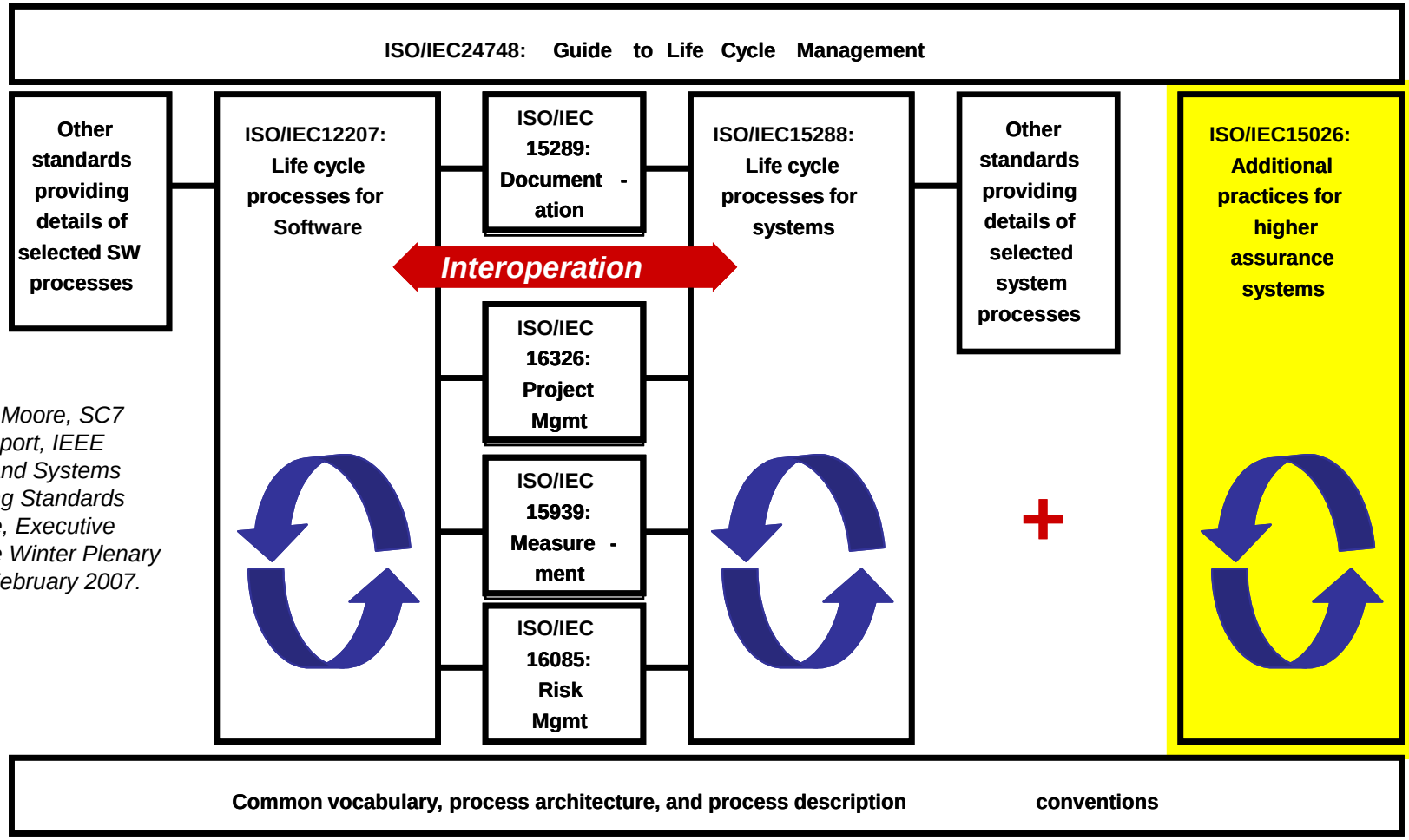
Table 1 –SwA Concern Categories -- (with interests relevant to security and privacy)		
SwA Concern Categories	**Risks**	**Purpose for Questions**
Service Confidentiality Policies	Without policies to enforce client data confidentiality/ privacy, acquirer's data could be at risk without service supplier liability.	To determine the service provider's confidentiality and privacy policies and ensure their enforcement.

Table 3 - Questions for Hosted Applications	
No.	Questions
Service Confidentiality Policies	
1	What are the customer confidentiality policies? How are they enforced?
2	What are the customer privacy policies? How are they enforced?
3	What are the policies and procedures used to protect sensitive information from unauthorized access? How are the policies enforced?
4	What are the set of controls to ensure separation of data and security information between different customers that are physically located in the same data center? On the same host server?
Operating Environment for Services	
5	Who configures and deploys the servers? Are the configuration procedures available for review, including documentation for all registry settings?
7	What are the data backup policies and procedures? How frequently are the backup procedures verified?
11	What are the agents or scripts executing on servers of hosted applications? Are there procedures for reviewing the security of these scripts or agents?
12	What are the procedures and policies used to approve, grant, monitor and revoke access to the servers? Are audit logs maintained?
13	What are the procedures and policies for handling and destroying sensitive data on electronic and printed media?
15	What are the procedures used to approve, grant, monitor, and revoke file permissions for production data and executable code?

We are engaged with many parts of the Community for Software Assurance-related standardization



ISO/IEC/IEEE 15026, System and Software Assurance



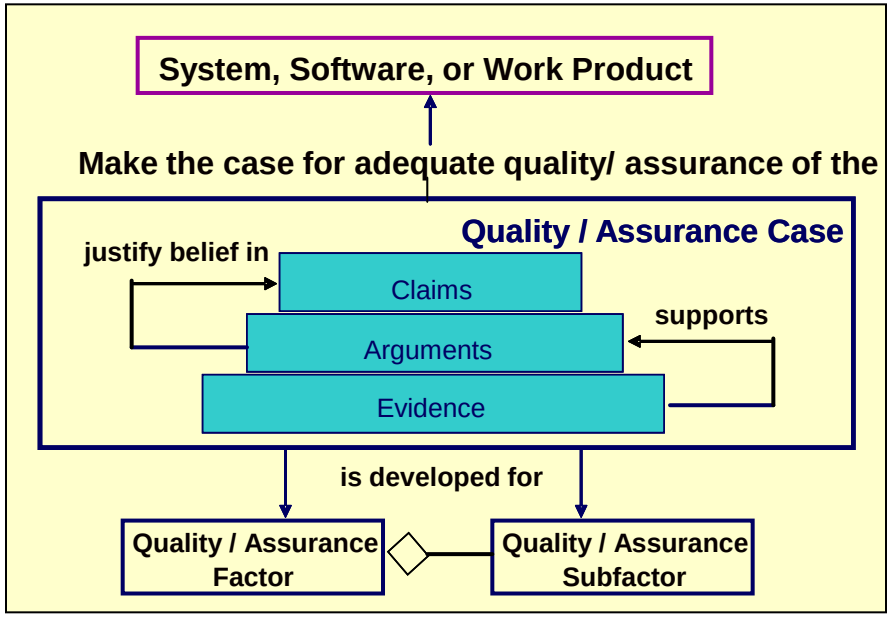
Source: J. Moore, SC7 Liaison Report, IEEE Software and Systems Engineering Standards Committee, Executive Committee Winter Plenary Meeting, February 2007.

"System and software assurance focuses on the management of risk and assurance of safety, security, and dependability within the context of system and software life cycle

Terms of Reference changed: ISO/IEC JTC1/SC7 WG7, previously "System and Software Integrity" SC7 WG9

ISO/IEC/IEEE 15026 Assurance Case

- **Set of structured assurance claims, supported by evidence and reasoning (arguments), that demonstrates how assurance needs have been satisfied.**
  - Shows compliance with assurance objectives
  - Provides an argument for the safety and security of the product or service.
  - Built, collected, and maintained throughout the life cycle
  - Derived from multiple sources



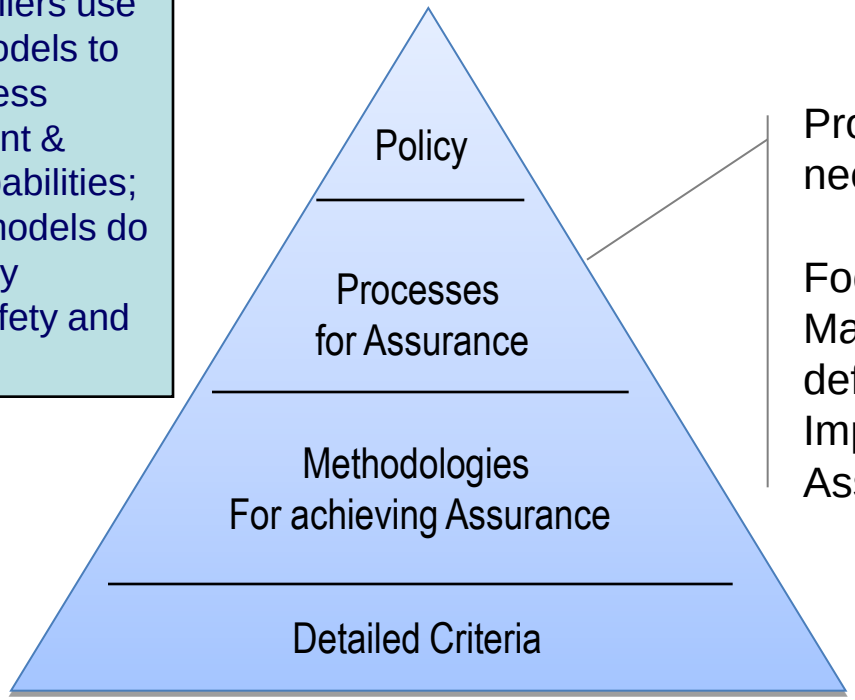
- **Sub-parts**
  - A high level summary
  - Justification that product or service is acceptably safe, secure, or dependable
  - Rationale for claiming a specified level of safety and security
  - Conformance with relevant standards & regulatory requirements
  - The configuration baseline
  - Identified hazards and threats and residual risk of each hazard / threat
  - Operational & support assumptions

Attributes

- Clear
- Consistent
- Complete
- Comprehensible
- Defensible
- Bounded
- Addresses all life cycle stages

Assurance in Maturity Models for Guiding Process Improvement

Many suppliers use maturity models to guide process improvement & assess capabilities; yet many models do not explicitly address safety and security.



Project leadership and team members need to know where and how to contribute

Focus Topic: Assurance for Capability Maturity Model Integration (CMMI)® defines the Assurance Thread for Implementation and Improvement of Assurance Practices

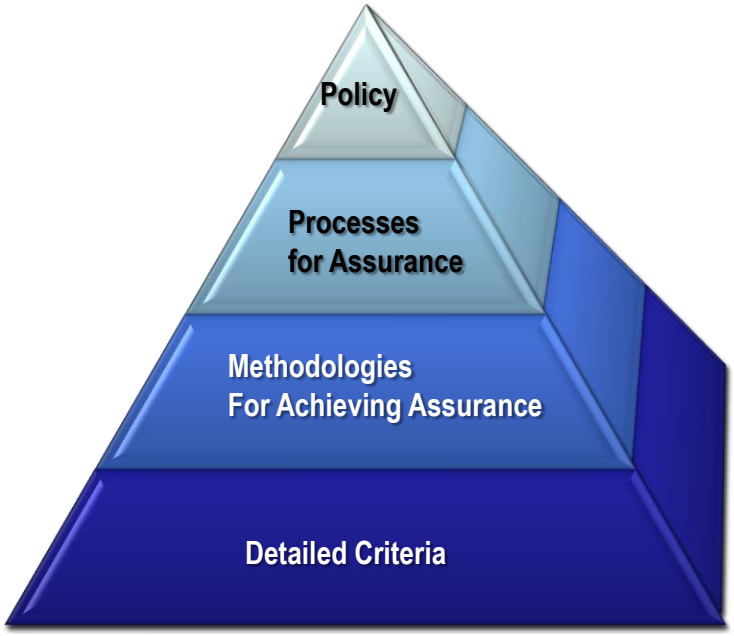
® Capability Maturity Model, Capability Maturity Modeling, and CMM are registered in the U.S. Patent & Trademark Office.

https://buildsecurityin.us-cert.gov/swa/procresrc.html

Experience gained for "Assurance" enhanced processes in *U.S. DoD and FAA joint project on Safety and Security Extensions for Integrated Capability Maturity Models, September 2004,* at SwA Community Resources and Information Clearinghouse - see https://buildsecurityin.us-cert.gov/swa/downloads/SafetyandSecurityExt-Sep2004.pdf

Other Assurance Maturity Models have been released in 2009:
The Building Security In Maturity Model (BSIMM) helps organizations plan software security initiatives http://www.bsi-mm.com/
The Software Assurance Maturity Model (SAMM) which is an open framework to help organizations formulate and implement a strategy for software security that is tailored to specific risks facing the organization http://www.opensamm.org/

Our Assurance Capability Framework Enables Communication



Project leadership and team members need to know where and how to contribute

- Assurance PRM defines the goals and practices needed to achieve SwA
- Assurance for CMMI ® defines the Assurance Thread for Implementation and Improvement of Assurance Practices that are assumed when using the CMMI-DEV

Understanding gaps helps suppliers and acquirers prioritize organizational efforts and funding to implement improvement actions

https://buildsecurityin.us-cert.gov/swa/procresrc.html

SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

Processes & Practices Goals

- Capture and discuss community of practices software assurance issues
- Share best practices
- Provide community input to and comments on:
  - DHS and DoD Guidebooks relating to Software Assurance
  - National and International Software Assurance Standards
  - DHS and DoD Policy Guidance on System and Software Assurance

Homeland Security

Processes & Practices Expected Outcomes

- In support of acquisition, management, and engineering and practices for software and systems assurance:
  - Community consensus standards for addressing assurance concerns throughout the system and software life cycles
  - Process benchmarking tools for assessing organizational capability with respect to assurance
  - Practice guidebooks providing compendiums of best practices and lessons learned
  - Community input to acquisition policy and guidance

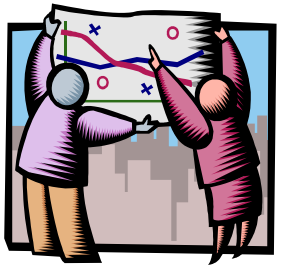
SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

Process Improvement Lifecycle - A Process for Achieving Assurance

Mission/Business Process

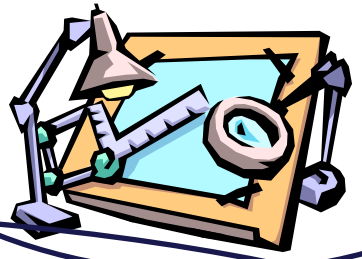
Understand Your Business Requirements for Assurance



Measure Your Results

Information System

Build or Refine and Execute Your Assurance Processes



Understand Assurance-Related Process Capability Expectations

Organization Support

Look to Standards for Assurance Process Detail

Adapted from: Paul Croll, Computer Sciences Corporation, August 2007

Analysis Of Common Practices

- Analyzed freely available models to determine how various models address similar goals and practices
- Identified the intersections of the common practices amongst the models regardless of the intended audience and levels of granularity
- Intended to support “Getting Started” by increasing awareness of improving software assurance by:
  - Learning how multiple models address similar assurance goals
  - Selecting practices from these models
- Provides a means for selecting models and practices that are best suited for the individual needs of various organizations

SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

Common SwA References Recommendations for Training

Assurance PRM	SAFEcode	MS SDL	Open SAMM	BSIMM
•Establish and maintain the strategic assurance training needs of the organization •Ensure resources have the training needed to do their job	1. Foundational (everyone) 2. Advanced (secure coding and testing practices) 3. Specialized (role-based)	1. Basic Concepts 2. Common Baseline 3. Custom Training	1. Technical Security Awareness training 2. Role specific guidance 3. Comprehensive security training and certifications	1. Create the software security satellite 2. Make customized, role-based training available on demand 3. Provide recognition for skills and career path progression

Source: SwA Benchmarking and Implementation, Moss, SSTC 2010

Objectives for Creating A (Self) Assessment Tool

- Organizations must be able to understand and become aware of risk throughout the supply chain.
  - What assurance goals are being met?
  - What practices are being implemented?
  - Who are the suppliers and how are they managing risk?
- Organizations need to be able to quantify and baseline assurance and risk management activities to ensure rugged software and software services are being developed and acquired.
- Supply chain partners must achieve increased awareness and communication to effectively understand risk throughout the software supply chain.

SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

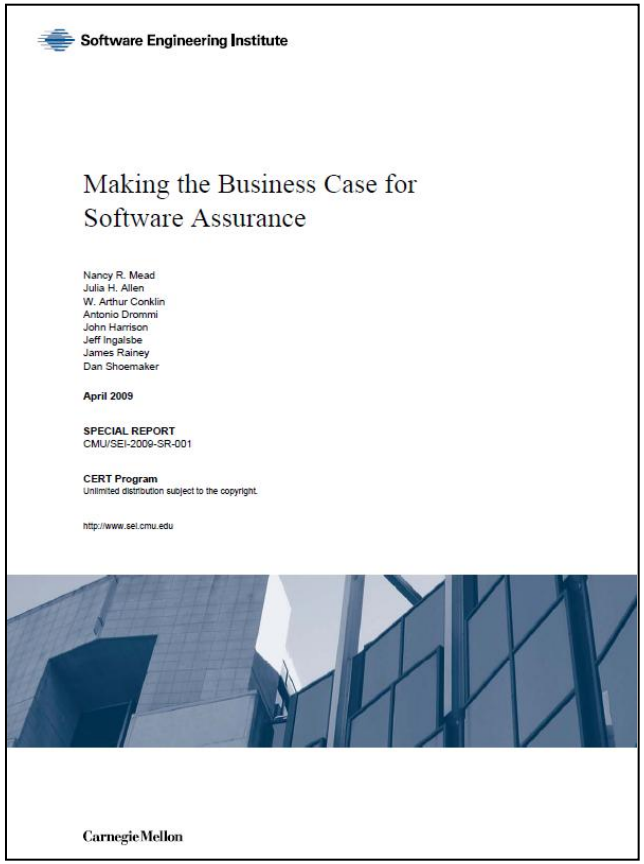
SwA Self-Assessment (High Level)

Role	Goal	Expected Practice	Activities	Source	BSIMM	CMMI-ACQ	OSAM M	RMM	MS SDL	Developer Considerations	Acquirer Considerations	Practice Implementation Level	Notes
DEV	SG 3.1 Establish assurance requirements.	SP 3.1.1 Understand the operating environment and define the operating constraints for assurance within the environments of system deployment.	Identify the system assurance context. Identify the system vulnerabilities with each operating environment defined for the system. Identify applicable assurance laws, policies, and constraints.	AF RD SP 1.1		PP SG1	EH1A	EF SG1 - SG2					
		SP 3.1.2 Develop customer assurance requirements.		AF RD SP 1.2	SR1.1	ARD SG1, SG3	SR1A	RRD:SG1 - SG3					
					SR1.2	REQM SG1	SR1B	COMP:SG 2					
					SR1.3		SR2A	KIM:SG6					
					SR2.3		SR2B	RRM:SG1					
		SP 3.1.3 Define product and product component assurance requirements		AF SP 2.1	SFD3.2	CM SG1	SA3A	KIM:SG2	P7				
							SA3B		P2				
		SP 3.1.4 Identify operational concepts and associated scenarios for intended and unintended use and associated assurance considerations.		AF RD SP3.1	AM1.1	RSKM SG1 - SG2	TA1A	RISK:SG1 - SG4					
					AM1.3		TA1B	KIM:SG6					
					AM1.4		TA2A						
					AM2.1								
					AM2.2								
		SP 3.1.5 Analyze assurance requirements.	Ensure established assurance requirements for the product flow to lower level solutions. Verify requirements against assurance objectives	AF RD SP 3.5	SR1.3	ARD SG3	SR1B	RRD:SG3					
		SP 3.1.6 Balance assurance needs against cost benefits.		AF SP 3.4	SM1.5	ARD SG3	SM3A - SM3B	FRM:SG4 - SG5, RRD:SG3					
		SP 3.1.7 Obtain Agreement of risk for Assurance level.			SM2.4	RSKM SG2	SM1A	RISK SG4, KIM SG3					
DEV	SG 3.2 Architect a solution for assurance.	SP 3.2.1 Develop alternative solutions and selection criteria for assurance.	Identify assurance defects and effectiveness of corrective actions in relevant products/systems/operations and apply lessons learned to alternative solutions; Understand the assurance capabilities of other products similar to the one under development that have been developed	TS SP 1.1	SFD1.1	ATM SG2	SA1A	RTSE:SG1 - SG2					
					SFD1.2	AVAL SG2	SA1B	KIM:SG2, SG6					
		SP 3.2.2 Architect for assurance.	Ensure the assurance of the product from the end-user's perspective; Ensure the customer's assurance responsibilities are specified; Identify resources and trust	AF TS SP2.1	SFD2.1	ATM SG2	SA2A	RTSE:SG3	P7				
					SFD2.3	AVAL SG2	SA2B						
		SP 3.2.3 Design for assurance.	Understand threat related design issues for design alternatives Emphasize potential design issues related to threat models or risk scenarios when considering design	AF TS SP 2.1	SFD2.1				P7				
		SP 3.2.4 Implement the assurance designs of the product components.		AF TS SP 3.1	AA3.2		SA1B						
		SP 3.2.5 Identify deviations from assurance coding standards. Implement appropriate mitigation to meet defined assurance objectives.		AF TS SP 3.1	CR1.4	AVER SG3	CR2A	RTSE:SG2					
					CR2.3		CR2B	RTSE:SG3					
					CR3.1		CR3A						

Business Case for Software Assurance

April 2009 SwA Report provides background, context and examples:

- Motivators
- Cost/Benefit Models Overview
- Measurement
- Risk
- Prioritization
- Process Improvement & Secure Software
- Globalization
- Organizational Development
- Case Studies and Examples


Software Engineering Institute

Making the Business Case for Software Assurance

Nancy R. Mead
Julia H. Allen
W. Arthur Conklin
Antonio Drommi
John Harrison
Jeff Ingalsbe
James Rainey
Dan Shoemaker

April 2009

SPECIAL REPORT
CMU/SEI-2009-SR-001

CERT Program
Unlimited distribution subject to the copyright.

http://www.sei.cmu.edu

Carnegie Mellon

SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

Security Measurement Resources

Oct 08 → Feb 09 → May 09 →

Practical Measurement Framework for Software Assurance and Information Security

Oct 2008

The Center for Internet Security

The CIS Security Metrics

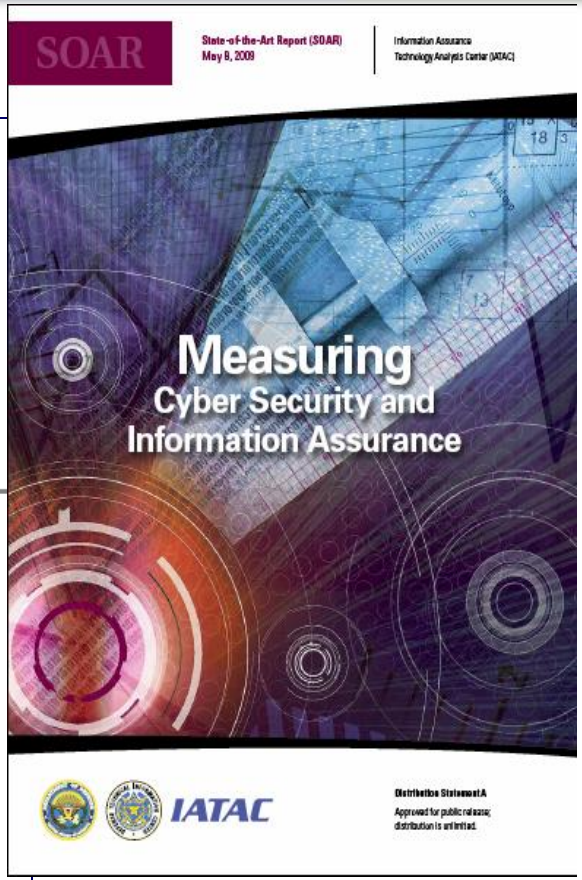
February 9

2009

Organizations struggle to make cost-effective security investment decisions; information security professionals lack widely accepted and unambiguous metrics for decision support. CIS established a consensus team of one hundred (100) industry experts to address this need. The result is a set of standard metric and data definitions that can be used across organizations to collect and analyze data on security process performance and outcomes.

This document contains twenty-one (21) metric definitions for six (6) important business functions: Incident Management, Vulnerability Management, Patch Management, Application Security, Configuration Management and Financial Metrics. Additional consensus metrics are currently being defined for these and additional business functions.

Consensus Metric Definitions



Measurement Guidance: Purpose

- To provide a practical framework for measuring software assurance achievement of SwA goals and objectives within the context of individual projects, programs, or enterprises.
  - Making informed decisions in the software development lifecycle related to information security compliance, performance, and functional requirements/controls
  - Facilitate adoption of secure software design practices
  - Mitigate risks throughout the System Development Lifecycle (SDLC) and ultimately reduce the numbers of vulnerabilities introduced into software code during development
  - Determining if security/performance/trade-offs have been defined and accepted
  - Assessing the trustworthiness of a system.
- Can be applied beyond SwA to a variety of security-related measurement efforts to help facilitate risk-based decision making through providing quantitative information on a variety of aspects of organization's security related performance.

Measurement Guidance: Scope & Resources

- Common measurement framework and measurement process leverage established measurement methodologies or emerging measurement methodologies that enjoy broad industry support:
  - NIST SP 800-55, *Security Metrics Guide for Information Technology Systems*
  - ISO/IEC 27004, *Information Security Management Measurement*
  - ISO/IEC 15939, *Software Engineering - Software Measurement Process*, also known as Practical Software and System Measurement (PSM)
  - Capability Maturity Model Integration (CMMI) Measurement & Analysis
  - CMMI Goal Question Indicator Measure (GQ(I)M)
- A listing of resources has been published on the SwA web site targeting primary stakeholder groups: Executive, Developer/Vendor/Supplier, Buyer/Acquirer
  - Sample SwA goals and questions lists to be used to define measures
  - Sources of measurable requirements, such as NIST documents
  - Articles on related subjects, including SwA measurement, security measurement, and software security measurement
  - Useful links
  - Measures library

Homeland Security

Software Assurance Ecosystem: The Formal Framework

The value of formalization extends beyond software systems to include related software system process, people and documentation

Process Docs & Artifacts

Requirements/Design Docs & Artifacts

Reports
Risk Analysis, etc)

Process, People & Documentation Evaluation Environment

- Some point tools to assist evaluators but mainly manual work
- Claims in Formal SBVR vocabulary
- Evidence in Formal SBVR vocabulary
- Large scope requires large effort

Process, People, documentation Evidence

Formalized Specifications

Claims, Arguments and Evidence Repository

- Formalized in SBVR vocabulary
- Automated verification of claims against evidence
- Highly automated and sophisticated risk assessments using transitive inter-evidence point relationships

Software System / Architecture Evaluation

- Many integrated & highly automated tools to assist evaluators
- Claims and Evidence in Formal vocabulary
- Combination of tools and ISO/OMG standards
- Standardized SW System Representation In KDM
- Large scope capable (system of systems)
- Iterative extraction and analysis for rules

Software system Technical Evidence

Executable Specifications

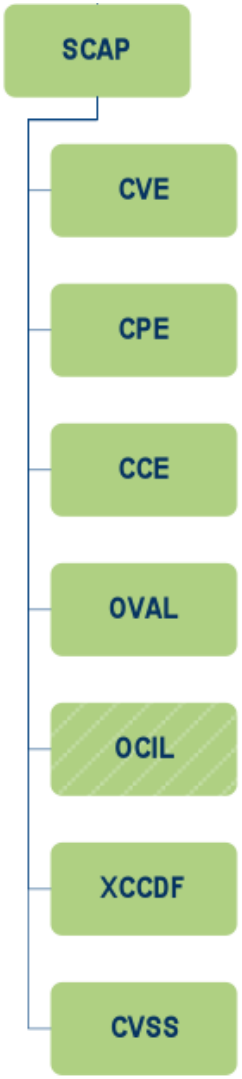
Software System Artifacts

Hardware Environment

Protection Profiles

IA Controls

CWE



SCAP 1.1 uses the following specifications:

- Extensible Configuration Checklist Description Format (XCCDF) 1.1.4, a language for authoring security checklists/benchmarks and for reporting results of checklist evaluation [QUI08]
- Open Vulnerability and Assessment Language (OVAL) 5.6, a language for representing system configuration information, assessing machine state, and reporting assessment results
- Open Checklist Interactive Language (OCIL) 2.0, a language for representing security checks that requires human feedback
- Common Platform Enumeration (CPE) 2.2, a nomenclature and dictionary of hardware, operating systems, and applications [BUT09]
- Common Configuration Enumeration (CCE) 5, a nomenclature and
  configurations
- Common Vulnerabilities and Exposures (CVE), a nomenclature an
  software flaws[9]
- Common Vulnerability Scoring System (CVSS) 2.0, an open speci
  severity of software flaw vulnerabilities [MEL07].

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

Special Publication 800-126
Revision 1 (DRAFT)

The Technical Specification for the Security Content Automation Protocol (SCAP): SCAP Version 1.1 (DRAFT)

Recommendations of the National Institute of Standards and Technology

Stephen Quinn
David Waltermire
Christopher Johnson
Karen Scarfone
John Banghart

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

Special Publication 800-126
Revision 1 (DRAFT)

The Technical Specification for the Security Content Automation Protocol (SCAP): SCAP Version 1.1 (DRAFT)

Recommendations of the National Institute of Standards and Technology

Stephen Quinn
David Waltermire
Christopher Johnson
Karen Scarfone
John Banghart

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

Special Publication 800-126
Revision 1 (DRAFT)

The Technical Specification for the Security Content Automation Protocol (SCAP): SCAP Version 1.1 (DRAFT)

Recommendations of the National Institute of Standards and Technology

Stephen Quinn
David Waltermire
Christopher Johnson
Karen Scarfone
John Banghart

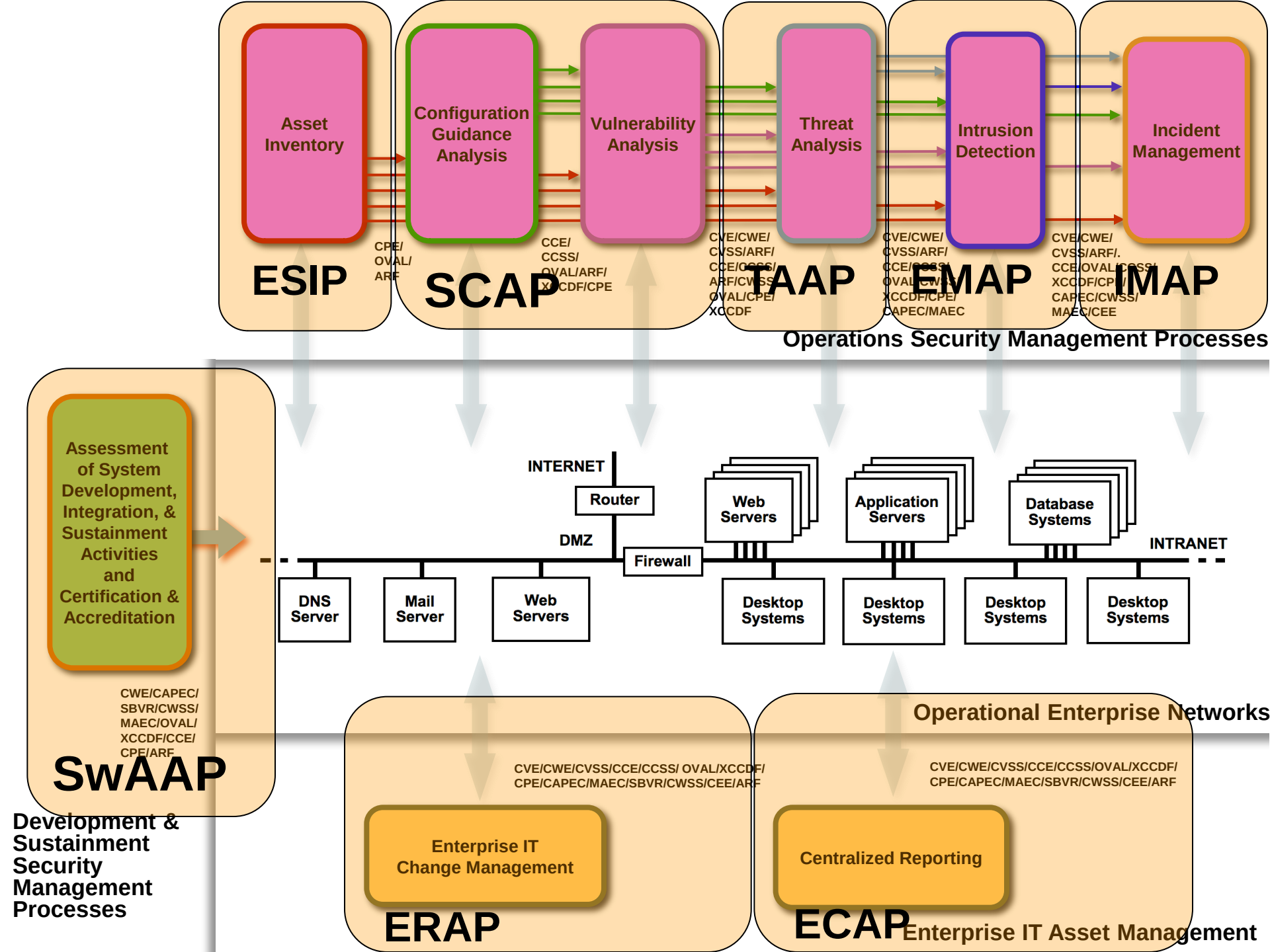
- **Software Assurance Automation Protocol (SwAAP)**
  - For measuring & enumerating software weaknesses and the assurance cases.

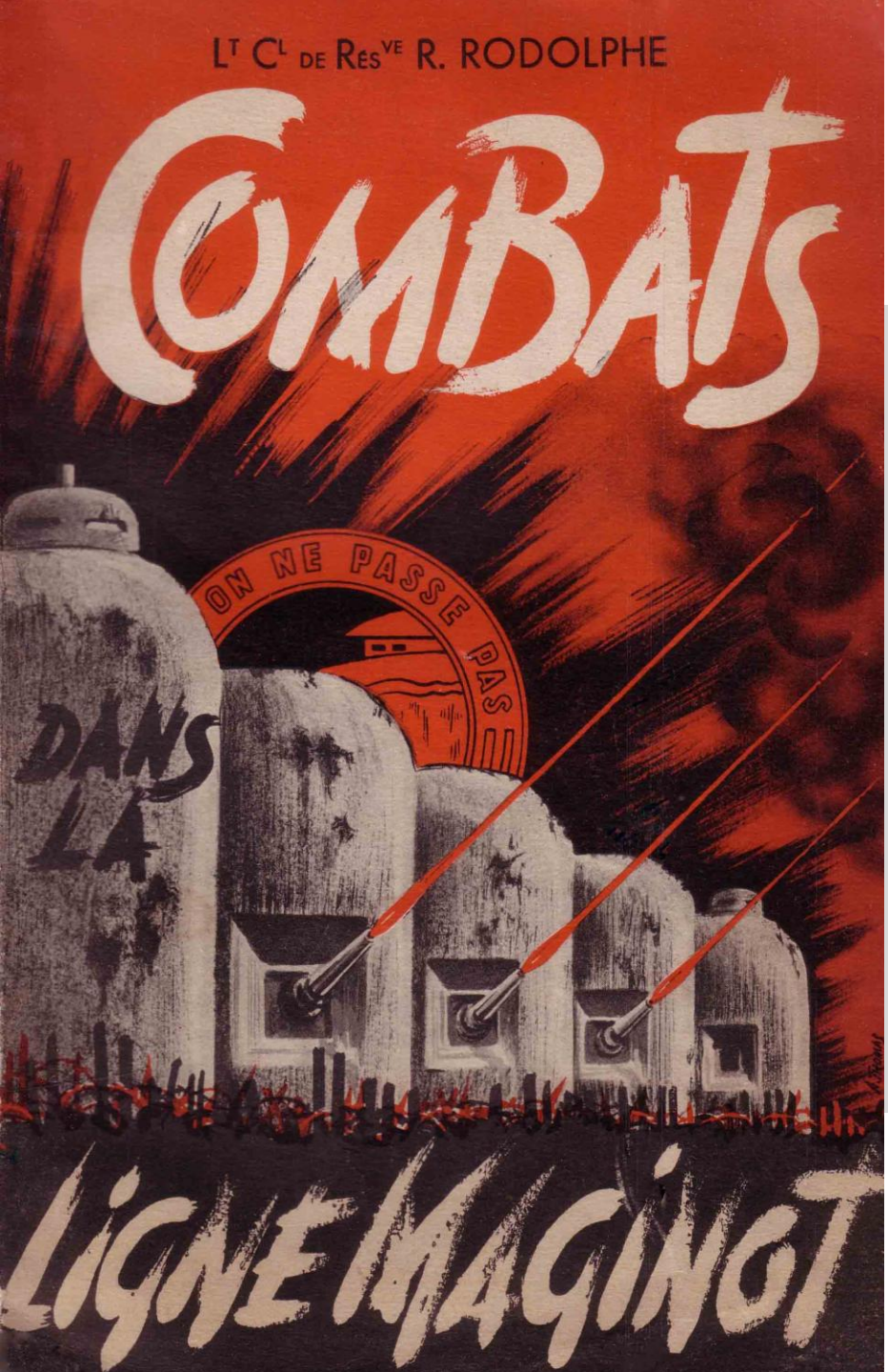
Common Weakness Enumeration (CWE),
Common Attack Pattern Enumeration & Classification (CAPEC),
Malware Attribute Enumeration & Characterization (MAEC),
Common Weakness Scoring System (CWSS),
Software Assurance Findings Expression Schema (SAFES),
NIST SAMATE's "Software Transparency Label",
ISO/IEC 15026 "Assurance Case" (ISO 15026),
OMG Software Assurance Evidence Metamodel (OMG SAEM),
OMG Argumentation Metamodel (OMG ARG),
OMG Structured Metrics Metamodel (OMG SMM),
OMG Knowledge Discovery Metamodel (OMG KDM),
OMG Abstract Syntax Tree Metamodel (OMG ASTM)

- plus SCAP to capture "accredited" system CPEs and CCE settings?
- OVAL checks for capturing "finger print" of software applications to address supply-chain risk measurement?

“Other” Automation Protocols (“O”AP)

- Event Management Automation Protocol (EMAP)
  - For reporting of security events.
  - Uses Common Event Expression (CEE), Malware Attribute Enumeration & Characterization (MAEC), CAPEC, etc.
- Enterprise Remediation Automation Protocol (ERAP)
  - For automated remediation of mis-configuration & missing patches.
  - Uses Common Remediation Enumeration (CRE) and Extended Remediation Information (ERI).
- Enterprise Compliance Automation Protocol (ECAP)
  - For reporting configuration compliance.
  - Uses Asset Reporting Format (ARF), Open Checklist Reporting Language (OCRL), etc.
- Enterprise System Information Protocol (ESIP)
  - For reporting of asset inventory information.
  - Uses …..
- Threat Analysis Automation Protocol (TAAP)
  - For analyzing threats and security risks.
  - Uses….
- Incident Management Automation Protocol (IMAP)
  - For supporting incident management and response.
  - Uses IODEF, etc


Asset Inventory
Configuration Guidance Analysis
Vulnerability Analysis
Threat Analysis
Intrusion Detection
Incident Management
CPE/ OVAL/ ARF
CCE/ CCSS/ OVAL/ARF/ XCCDF/CPE
CVE/CWE/ CVSS/ARF/ CCE/CCSS/ ARF/CWSS/ OVAL/CPE/ XCCDF
CVE/CWE/ CVSS/ARF/ CCE/CCSS/ OVAL/CWSS/ XCCDF/CPE/ CAPEC/MAEC
CVE/CWE/ CVSS/ARF/. CCE/OVAL/CCSS/ XCCDF/CPE/ CAPEC/CWSS/ MAEC/CEE
ESIP
SCAP
TAAP
EMAP
IMAP
Operations Security Management Processes
Assessment of System Development, Integration, & Sustainment Activities and Certification & Accreditation
CWE/CAPEC/ SBVR/CWSS/ MAEC/OVAL/ XCCDF/CCE/ CPE/ARF
SwAAP
INTERNET
Router
DMZ
Firewall
Web Servers
Application Servers
Database Systems
INTRANET
DNS Server
Mail Server
Web Servers
Desktop Systems
Desktop Systems
Desktop Systems
Desktop Systems
Operational Enterprise Networks
CVE/CWE/CVSS/CCE/CCSS/ OVAL/XCCDF/ CPE/CAPEC/MAEC/SBVR/CWSS/CEE/ARF
CVE/CWE/CVSS/CCE/CCSS/OVAL/XCCDF/ CPE/CAPEC/MAEC/SBVR/CWSS/CEE/ARF
Enterprise IT Change Management
Centralized Reporting
ERAP
ECAP
Enterprise IT Asset Management
Development & Sustainment Security Management Processes



Today's attack vector is through software

"A fortress mentality will not work in cyber. **We cannot retreat behind a Maginot Line of firewalls**…If we stand still for a minute, our adversaries will overtake us."

-William Lynn, U.S. Deputy Secretary of Defense
January 2010

The Rugged Software Manifesto

I am rugged... and more importantly, my code is rugged.

I recognize that software has become a foundation of our modern world.

I recognize the awesome responsibility that comes with this foundational role.

I recognize that my code will be used in ways I cannot anticipate, in ways it was not designed, and for longer than it was ever intended.

I recognize that my code will be attacked by talented and persistent adversaries who threaten our physical, economic, and national security.

I recognize these things - and I choose to be rugged.

I am rugged because I refuse to be a source of vulnerability or weakness.

I am rugged because I assure my code will support its mission.

I am rugged because my code can face these challenges and persist in spite of them.

I am rugged, not because it is easy, but because it is necessary... and I am up for the challenge.

The MANIFESTO

Focus on Resilience and Survivability - If compromised, damage to the software will be minimized, and it will recover quickly to an acceptable level of operating capacity; it is ‘rugged’

I am rugged - and more importantly, my code is rugged.

I recognize that software has become a foundation of our modern world.

I recognize the awesome responsibility that comes with this foundational role.

I recognize that my code will be used in ways I cannot anticipate, in ways it was not designed, and for longer than it was ever intended.

I recognize that my code will be attacked by talented and persistent adversaries who threaten our physical, economic, and national security.

I recognize these things - and I choose to be rugged.

I am rugged because I refuse to be a source of vulnerability or weakness.

I am rugged because I assure my code will support its mission.

I am rugged because my code can face these challenges and persist in spite of them.

I am rugged, not because it is easy, but because it is necessary… and I am up for the challenge.

The Rugged Software Manifesto

I am rugged... and more importantly, my code is rugged.

I recognize that software has become a foundation of our modern world.

I recognize the awesome responsibility that comes with this foundational role.

I recognize that my code will be used in ways I cannot anticipate, in ways it was not designed, and for longer than it was ever intended.

I recognize that my code will be attacked by talented and persistent adversaries who threaten our physical, economic, and national security.

I recognize these things - and I choose to be rugged.

I am rugged because I refuse to be a source of vulnerability or weakness.

I am rugged because I assure my code will support its mission.

I am rugged because my code can face these challenges and persist in spite of them.

I am rugged, not because it is easy, but because it is necessary... and I am up for the challenge.

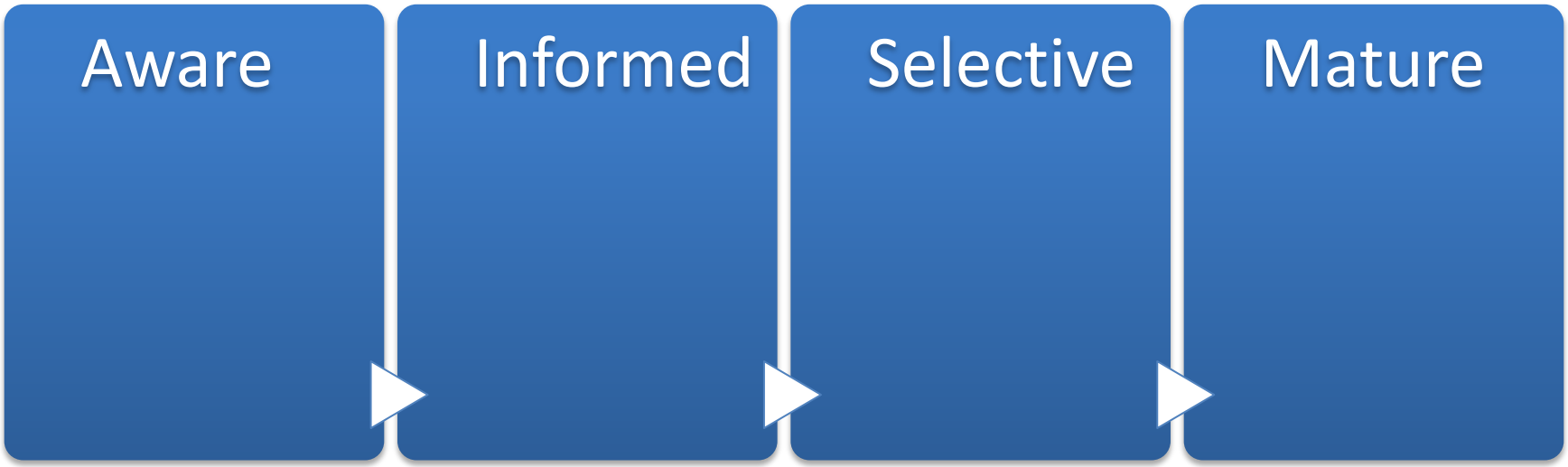
Rugged?

Twitter: @RuggedSoftware

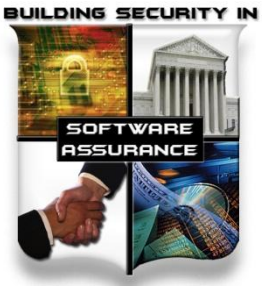
http://ruggedsoftware.org

{ RUGGED } software

The Journey



IT/Software Supply Chain Management is a National Security & Economic Issue

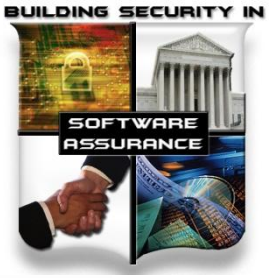


- Adversaries can gain “intimate access” to target systems, especially in a global supply chain that offers limited transparency
- Advances in science and technology will always outpace the ability of government and industry to react with new policies and standards
  - National security policies must conform with international laws and agreements while preserving a nation’s rights and freedoms, and protecting a nation’s self interests and economic goals
  - Forward-looking policies can adapt to the new world of global supply chains
  - International standards must mature to better address supply chain risk management, IT security, systems & software assurance
  - Assurance Rating Schemes for software products and organizations are needed
- IT/software suppliers and buyers can take more deliberate actions to security-enhance their processes and practices to mitigate risks
  - Government & Industry have significant leadership roles in solving this
  - Individuals can influence the way their organizations adopt security practices

Homeland Security

Globalization will not be reversed; this is how we conduct business – To remain relevant, standards and capability benchmarking measures must address “assurance” mechanisms needed to manage IT/Software Supply Chain risks.

Next SwA Forum 27 Sep – 1 Oct 2010 at NIST, Gaithersburg, MD



SOFTWARE ASSURANCE FORUM

"Building Security In"

https://buildsecurityin.us-cert.gov/swa

Joe Jarzombek, PMP, CSSLP
Director for Software Assurance
National Cyber Security Division
Department of Homeland Security
Joe.Jarzombek@dhs.gov
(703) 235-5126
LinkedIn SwA Mega-Community

Homeland Security

SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

Homeland Security

Commerce

National Defense

Next SwA Forum 27 Sep – 1 Oct 2010 at NIST, Gaithersburg, MD