SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

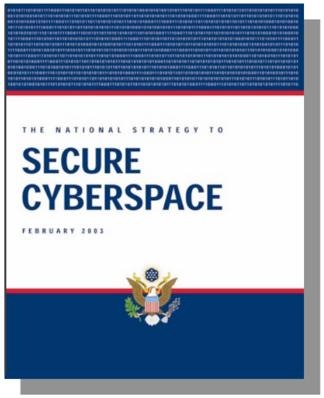
Homeland Security

Department of Commerce

National Defense

Next SwA Forum 9-12 March 2010 at MITRE, McLean VA

Next SwA Working Group Session 15-17 Dec 2009 at MITRE, McLean VA



THE NATIONAL STRATEGY TO

SECURE CYBERSPACE

FEBRUARY 2003

Software Assurance

A Strategic Initiative of the U.S. Department of Homeland Security to Promote Integrity, Security, and Reliability in Software

Collaboratively Advancing Strategies to Mitigate Software Supply Chain Risks

November 2009

Homeland Security

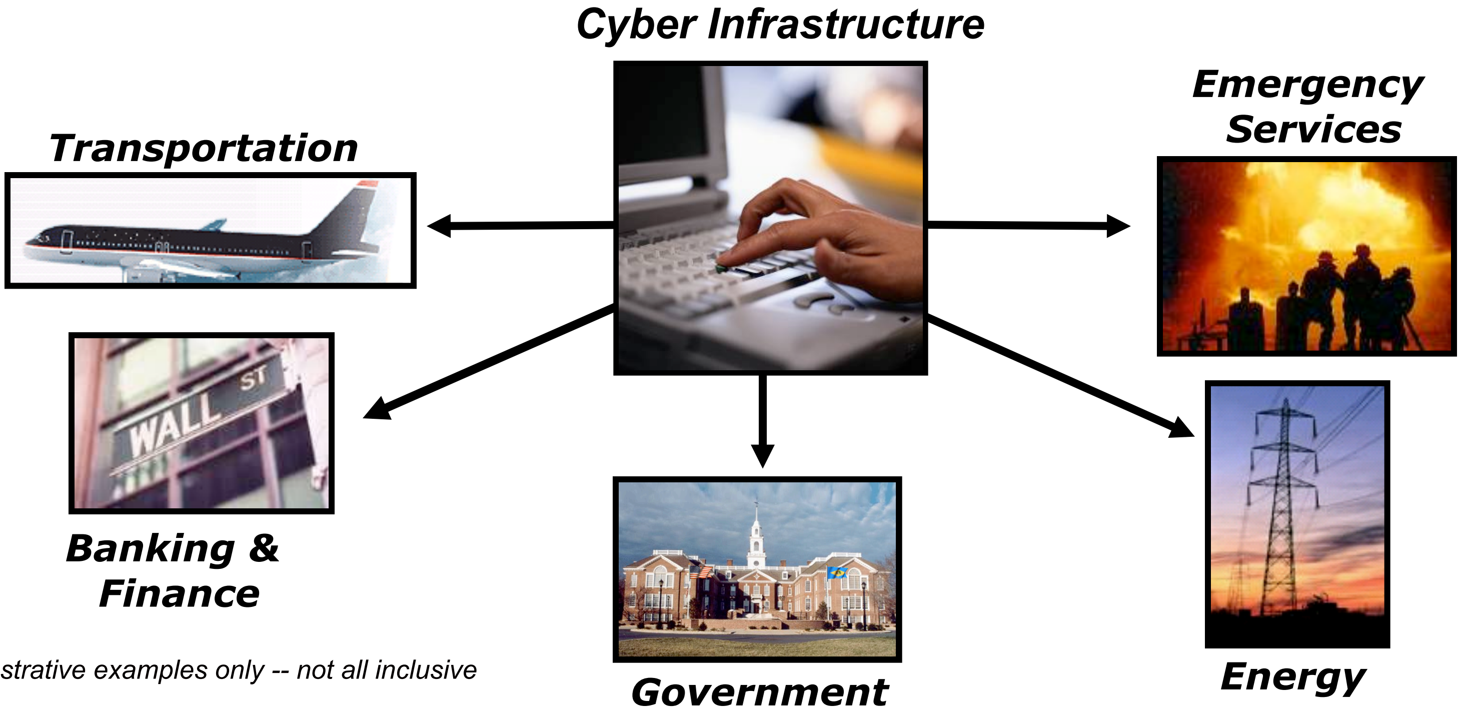
Joe Jarzombek, PMP, CSSLP
Director for Software Assurance
National Cyber Security Division
Office of the Assistant Secretary for
Cybersecurity and Communications

Cyberspace

- Cyberspace is composed of hundreds of thousands of globally interconnected computers, servers, routers, switches, and cables that allow the critical infrastructures to work.
  - It transcends physical, organizational and geopolitical boundaries and thus has global stakeholders from both the public and private sectors.
- It encompasses the logical layer where software applications, Web sites, bulletin boards, chat rooms, e-mail, and electronic exploits operate (e.g., viruses, Botnets, etc).
- While the Internet is part of cyberspace, it also includes the local and wide area networks, as well as the users connected to the Internet.
- These networks contain a wealth of information that includes proprietary, classified and privacy data and operate many of the nation’s critical infrastructure and key resources, to include the electrical Smart Grid.

Cyber Infrastructure: Critical to National and Economic Security

Cyber Infrastructure represents the convergence of information technology and communications systems, is inherent to nearly every aspect of modern life



Cyber Infrastructure

Transportation

Emergency Services

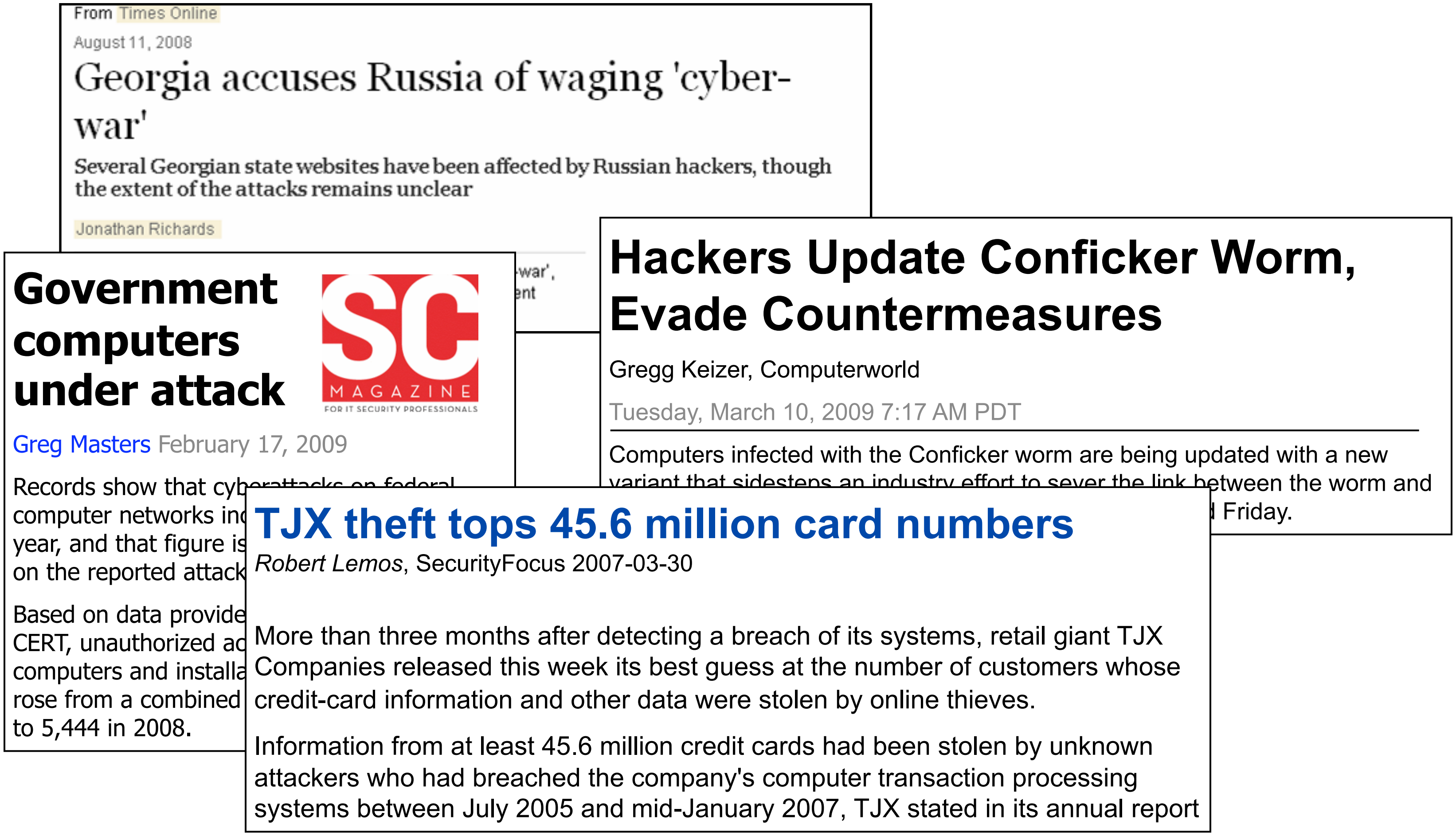
Banking & Finance

Government

Energy

Illustrative examples only -- not all inclusive

Cyber Incidents are Increasing in Frequency, Scale, and Sophistication



From Times Online

August 11, 2008

Georgia accuses Russia of waging 'cyber-war'

Several Georgian state websites have been affected by Russian hackers, though the extent of the attacks remains unclear

Jonathan Richards

-war',
ent

Government computers under attack

SC MAGAZINE
FOR IT SECURITY PROFESSIONALS

Greg Masters February 17, 2009

Records show that cyb
computer networks inc
year, and that figure is
on the reported attack

Based on data provide
CERT, unauthorized ac
computers and installa
rose from a combined
to 5,444 in 2008.

Hackers Update Conficker Worm, Evade Countermeasures

Gregg Keizer, Computerworld

Tuesday, March 10, 2009 7:17 AM PDT

Computers infected with the Conficker worm are being updated with a new variant that sidesteps an industry effort to sever the link between the worm and
Friday.

TJX theft tops 45.6 million card numbers

Robert Lemos, SecurityFocus 2007-03-30

More than three months after detecting a breach of its systems, retail giant TJX Companies released this week its best guess at the number of customers whose credit-card information and other data were stolen by online thieves.

Information from at least 45.6 million credit cards had been stolen by unknown attackers who had breached the company's computer transaction processing systems between July 2005 and mid-January 2007, TJX stated in its annual report

Defining the Threat to Cyber Networks and Systems

- Threats to cyber networks can come from numerous sources, including hostile governments, terrorist groups, disgruntled employees, and malicious intruders.
  - National Governments
  - Sub-national Terrorists Groups
  - Industrial Spies and Organized Crime Groups
  - Hacktivists
  - Hackers
- These threat actors employ an equally diverse collection of cyber tools that are generally easy to use, are difficult to attribute, and can have hard-to-predict and cascading impacts.

Defining the Threat to Cyber Networks and Systems

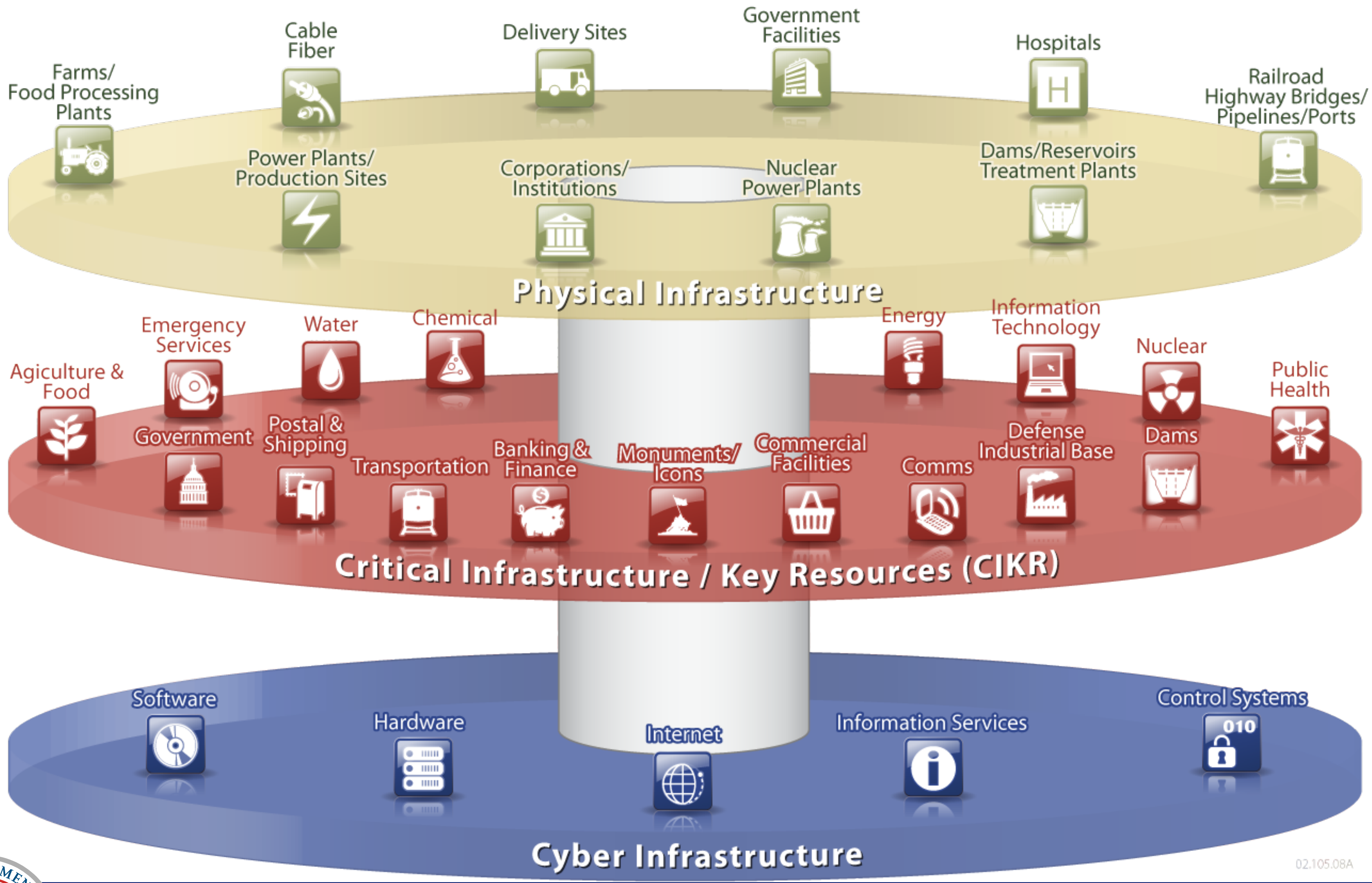
The Threat

- The threats are large and diverse, ranging from independent, unsophisticated, opportunistic hackers to very technically competent intruders and nation states using state-of-the-art intrusion techniques.
- Malicious actors are increasingly acquiring information technology skills to launch malicious attacks designed to steal information and disrupt, deny access to, degrade or destroy critical information and infrastructure systems.
- Hacker groups already possess the necessary skills to launch a successful cyber attack and may be “talent-for-hire” available to terrorist, criminal organizations, and nation states
- Attackers do not need to be technically savvy as free and commercial automated tools are simplifying attack methods
- Both actors and system vulnerabilities put infrastructure at risk.

Reliance on Cyberspace

- Society increasingly relies on technology and telecommunications to support our economy and business operations and critical functions of government
- Global wireless and cellular usage is on the rise
- To put individual demand in perspective,
  - 1.5 billion individuals currently utilize the Internet and this number is growing
  - Over 200 billion emails are sent per day
  - 8 hours of YouTube are uploaded every minute

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.

- **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.

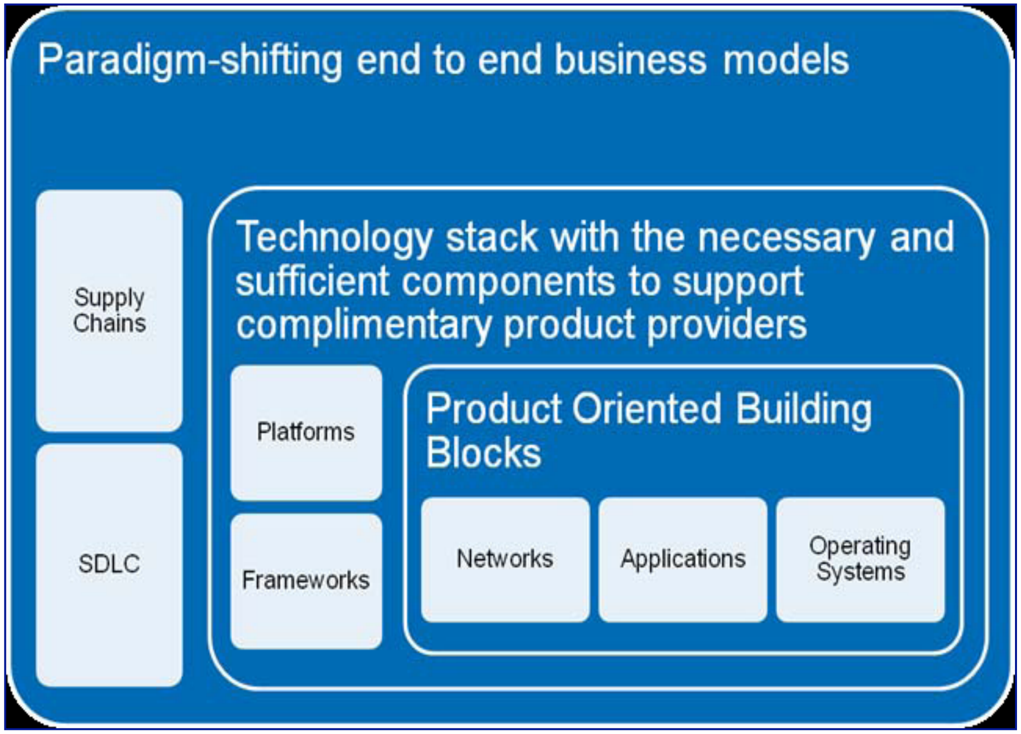
- **Enables software security automation and measurement capabilities through use of common indexing and reporting capabilities for malware, exploitable software weaknesses, and common attacks which target software.**
  - Collaborates with the National Institute of Standards and Technology, international 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).

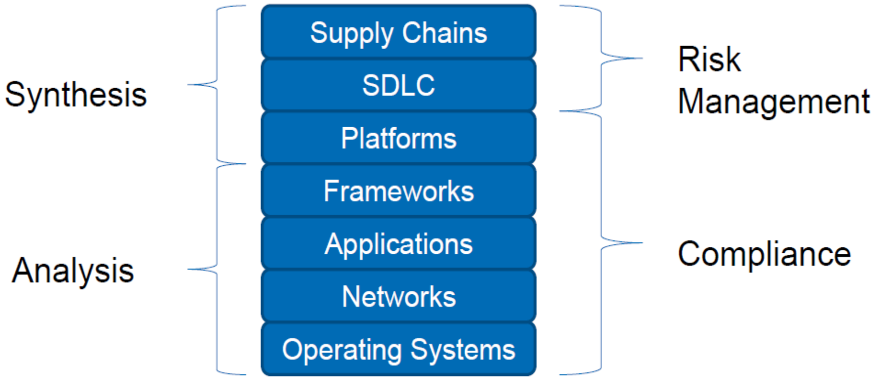
IT/software security risk landscape is a convergence between "defense in depth" and "defense in breadth"

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


Paradigm-shifting end to end business models
Supply Chains
SDLC
Technology stack with the necessary and sufficient components to support complimentary product providers
Platforms
Frameworks
Product Oriented Building Blocks
Networks
Applications
Operating Systems



Software Assurance provides a focus for:

-- Secure Software Components,

-- Security in the SDLC and

-- Software Supply Chain Risk Management

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

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

Recommendations Addressing Globalization of Software

Defense Science Board Task Force September 2007 Report on "Mission Impact of Foreign Influence on DoD Software"

Findings relate to:

-The Industry Situation

-Dependence on Software-

-Software Vulnerabilities

-Threat of the Nation-State Adversary

-Awareness of Software Assurance Threat and Risk

-Status of Software Assurance

-Ongoing Efforts in Software Assurance

-Supplier Trustworthiness Considerations

-Finding Malicious Code

-Government Access to Source Code

Recommendations relate to:

-Procurement of COTS and Off-Shore Software

-Increase US Insight into Capabilities and Intentions

-Offensive Strategies can complicate Defensive Strategies

-System Engineering and Architecture for Assurance

-Improve the Quality of Software

-Improve Tools and Technology for Assurance

-More Knowledgeable Acquisition of Software

-Research and Development in Software Assurance

Eliminate excess functionality in mission-critical components

Improve effectiveness of Common Criteria

Improve usefulness of assurance metrics

Promote use of automated tools in development

Increase transparency and knowledge of suppliers' processes

Components should be supplied by suppliers of commensurate trustworthiness

Custom code for critical systems should be developed by cleared US citizens

Provide incentives to industry to produce higher quality code; improve assuredness of COTS SW

Use risk-based acquisition

Research programs to advance vulnerability detection and mitigation

Advance the issue of software assurance and globalization on national agenda as part of effort to reduce national cyber risk

Assurance Challenges in Mitigating Software Supply Chain Risks

- Complexity hampers our ability to determine and predict code behavior; so any “assurance” claims for security/safety-critical applications are limited.
- Without adequate diagnostic capabilities and commonly recognized standards from which to assert claims about the assurance of products, systems and services, the “providence and pedigree of supply chain actors” become a more dominant consideration for security/safety-critical applications:
  - Consumers lack requisite transparency for more informed decision-making for mitigating risks;
  - Favoring domestic suppliers does not necessarily address ‘assurance’ in terms of capabilities to deliver secure/safe components.
- Several needs arise:
  - Need internationally recognized standards to support processes and provide transparency for more informed decision-making for mitigating enterprise risks.
  - Need ‘Assurance’ to be explicitly addressed in standards & capability benchmarking models for organizations involved with security/safety-critical applications.
  - Need more comprehensive diagnostic capabilities to provide sufficient evidence that “code behavior” can be well understood to not possess exploitable or malicious constructs.

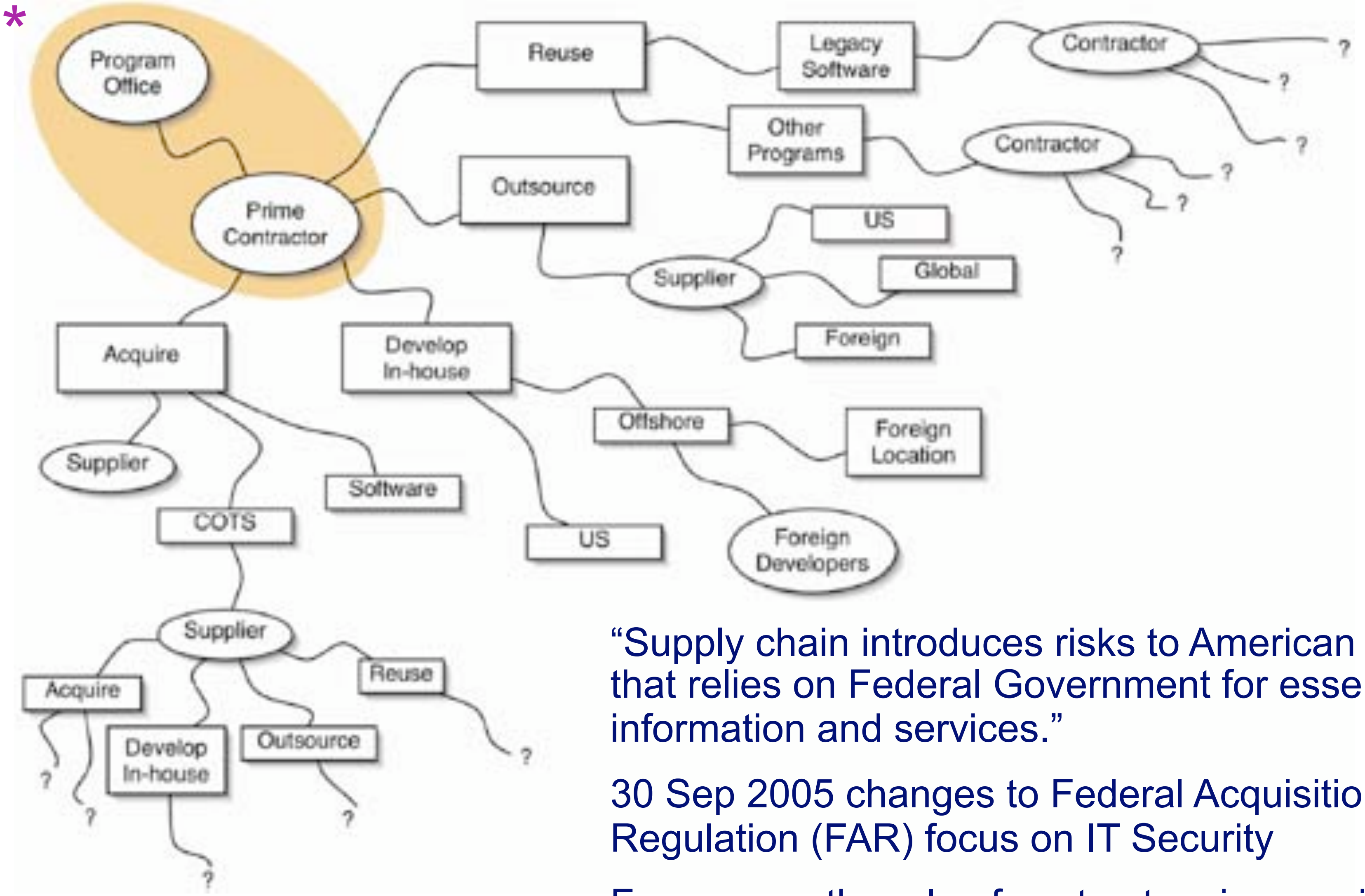
Security-Enhanced Capabilities:
Mitigating Risks to the Enterprise

- 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)

*



"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

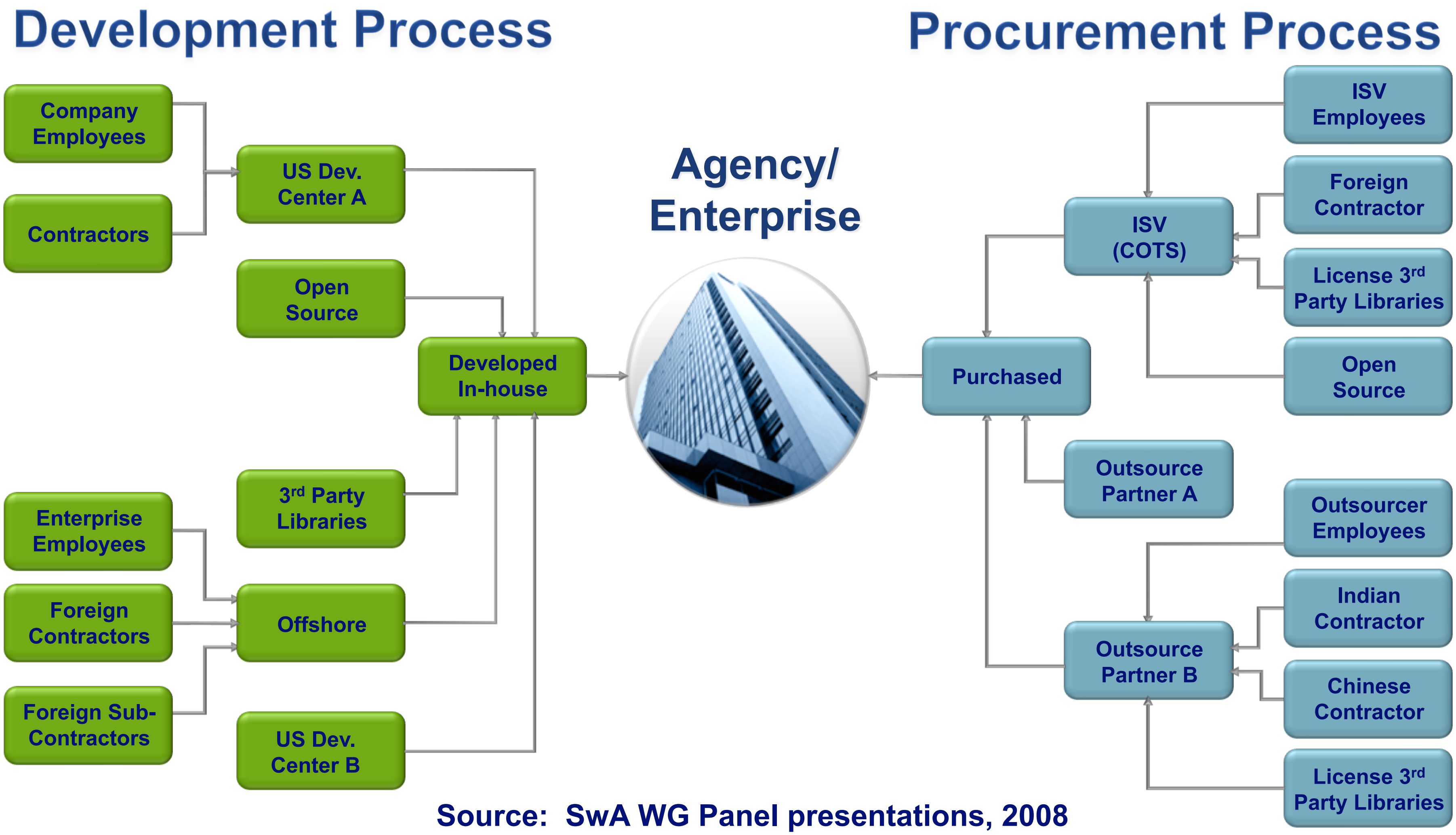
Focuses on the role of contractors in security as Federal agencies outsource various IT functions.

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

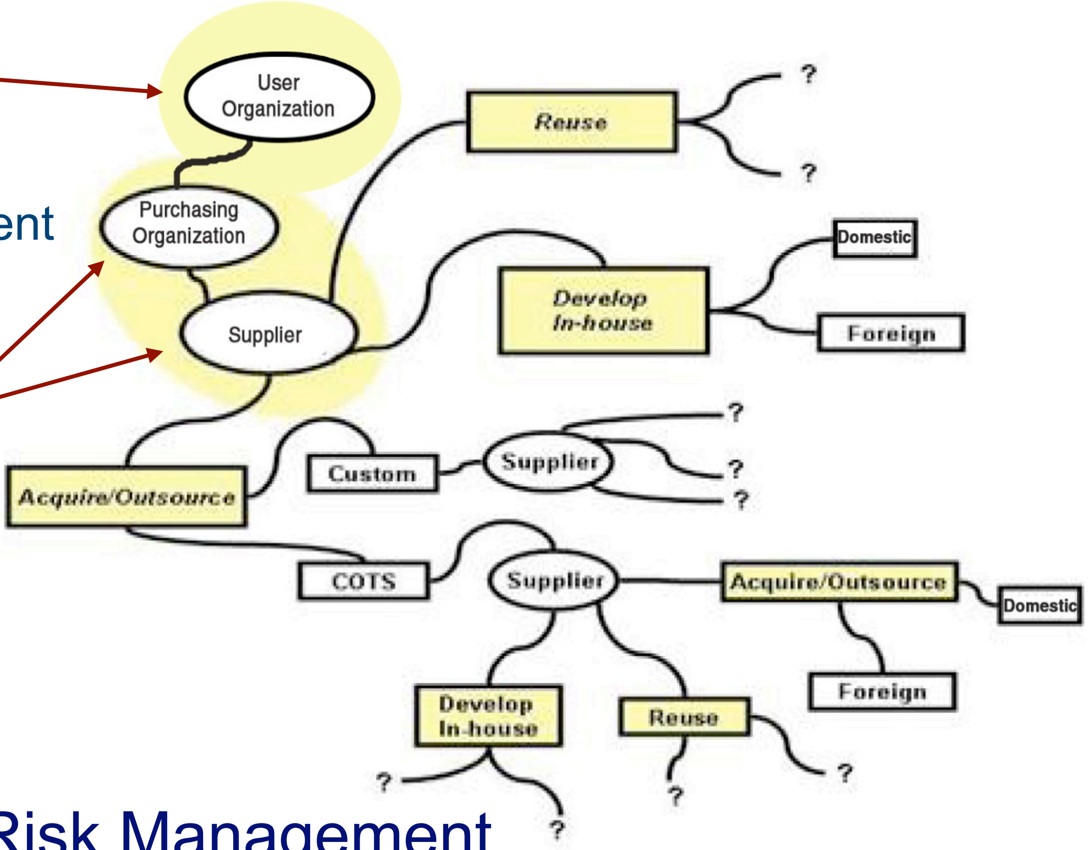
New Considerations for Quality & Security



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

Supply System Attacks

- Why send malicious code over the Internet if you can pre-infect computer parts or consumer devices?
- Some recent examples:
  - Fall 2007: hard drives from China arrived on store shelves pre-infected with a virus
  - Christmas 2007: hundreds of digital photo frames, USB memory sticks, GPS devices, and other plug-n-play devices were found to be infected with malware
  - January 2008: FBI announces a multi-year investigation into counterfeit Cisco routers
- **Exploitation potential of non-secure IT/software is often independent of "intent"**

Adopted in part from Marcus H. Sachs, Verizon, *"Supply Chain Risk Management: Can we Secure the IT Supply Chain in the Age of Globalization?" Software Assurance Forum, 15 Oct 2008*

Major pipelines for IT/Software Supply Chain

1. From country where manufactured
   - to a certified domestic distributor to domestic end-user, or
   - through a certified distributor in a second country to domestic end-user

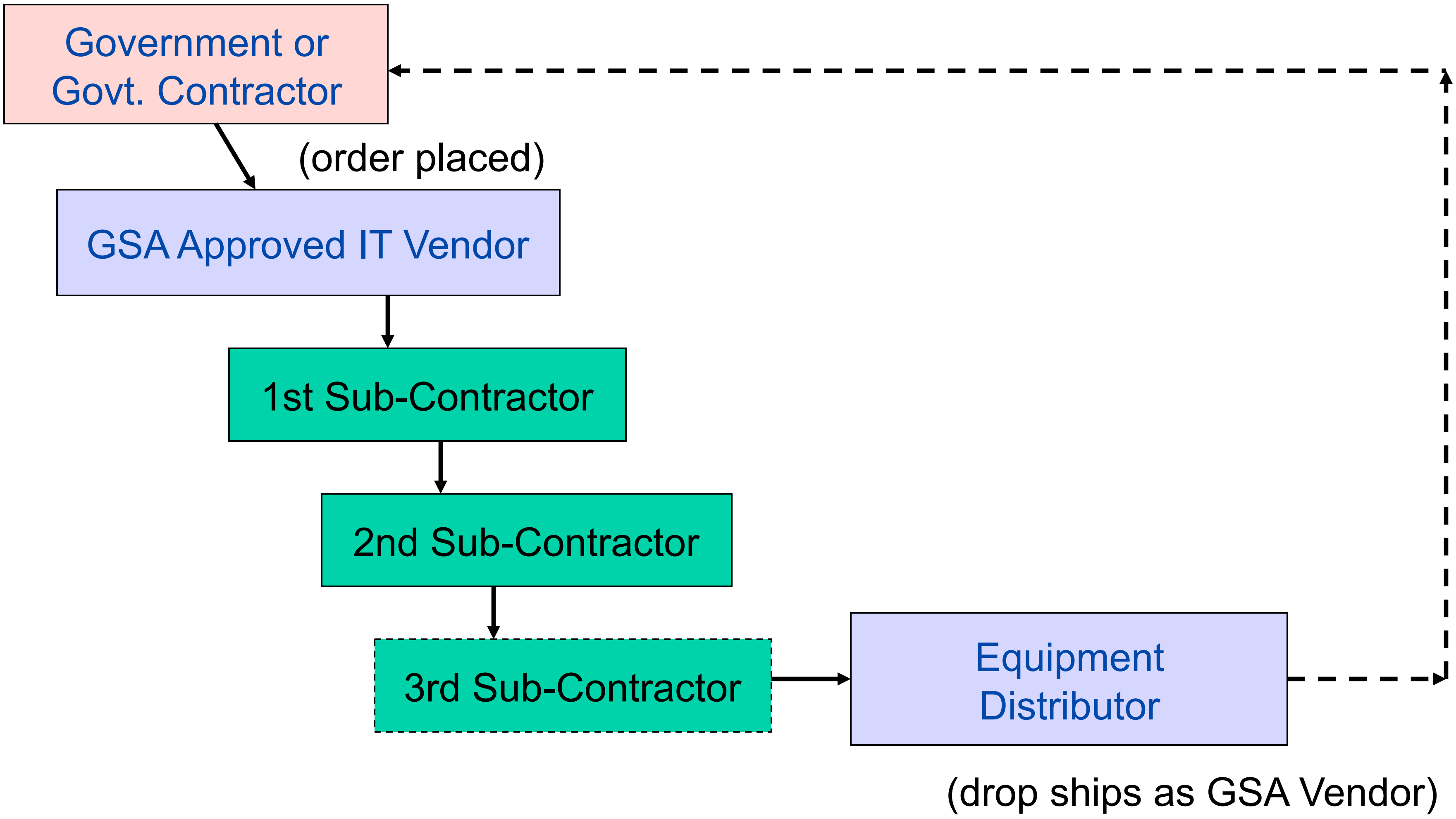
2. From country of origin
   - to online auction site (such as eBay or similar) to end-user
   - to distributor or retailer with unknown credentials to end-user

3. In most cases, IT/software is manufactured/produced by a non-vetted or uncertified supplier (especially for software) to domestic end-user

4. Transparency of supply chain complicated through re-supply of integrators, VARs, and service providers

Adopted in part from Marcus H. Sachs, Verizon, *"Supply Chain Risk Management: Can we Secure the IT Supply Chain in the Age of Globalization?" Software Assurance Forum, 15 Oct 2008*

US Government Contracting Process



The New Issue is Virtual Security

- In addition to physical security, we now worry about cyber risks:
  - Theft of intellectual property
  - Fake or counterfeit products
  - Import/export of strong encryption
  - IT/software with deliberately embedded malicious functionality
    - Logic bombs and self-modifying code
    - Other “added features” like key loggers
    - Deliberately hidden back doors for unauthorized remote access
  - Exploitable IT/software from suppliers with poor security practices
    - Failure to use manufacturing processes/capabilities to design and build secure products (no malicious intent) in delivering exploitable products
    - Resuppliers (VARs, integrators, and service providers) often lack incentives and capabilities to adequately check content of sub-contracted and outsourced IT/software products

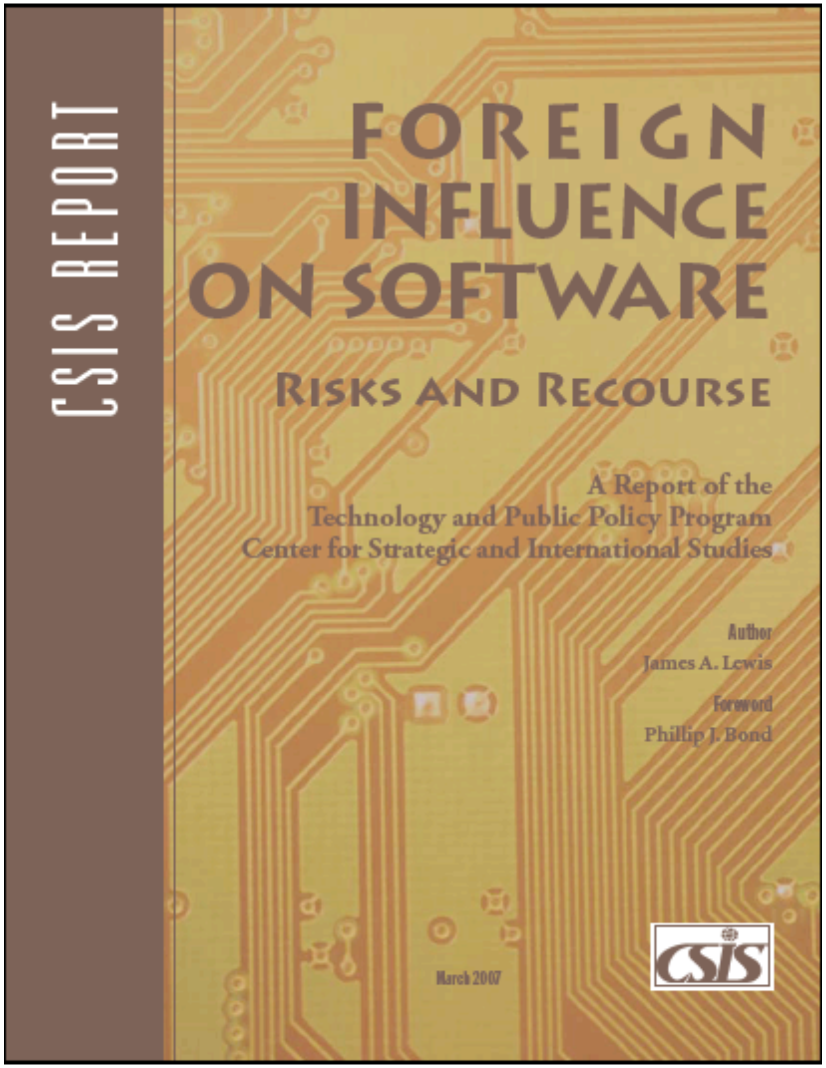
- IT/software security laws, policies, & standards are immature

Adopted in part from Marcus H. Sachs, Verizon, *”Supply Chain Risk Management: Can we Secure the IT Supply Chain in the Age of Globalization?” Software Assurance Forum, 15 Oct 2008*

Recommendations Addressing Globalization of Software

Center for Strategic and International Studies Report on Risks and Recourse

1. Assess risk (and share assessment)
2. Focus on assurance, not location
3. Avoid one-size-fits-all solutions
4. Refocus and reform existing certification processes
5. Identify commercial best practices and tools and expand their use
6. Create governance structure(s) for assurance
7. Accelerate info assurance efforts
8. Promote leadership in IT innovation



CSIS REPORT

FOREIGN INFLUENCE ON SOFTWARE

RISKS AND RECOURSE

A Report of the
Technology and Public Policy Program
Center for Strategic and International Studies

Author
James A. Lewis

Foreword
Phillip J. Bond

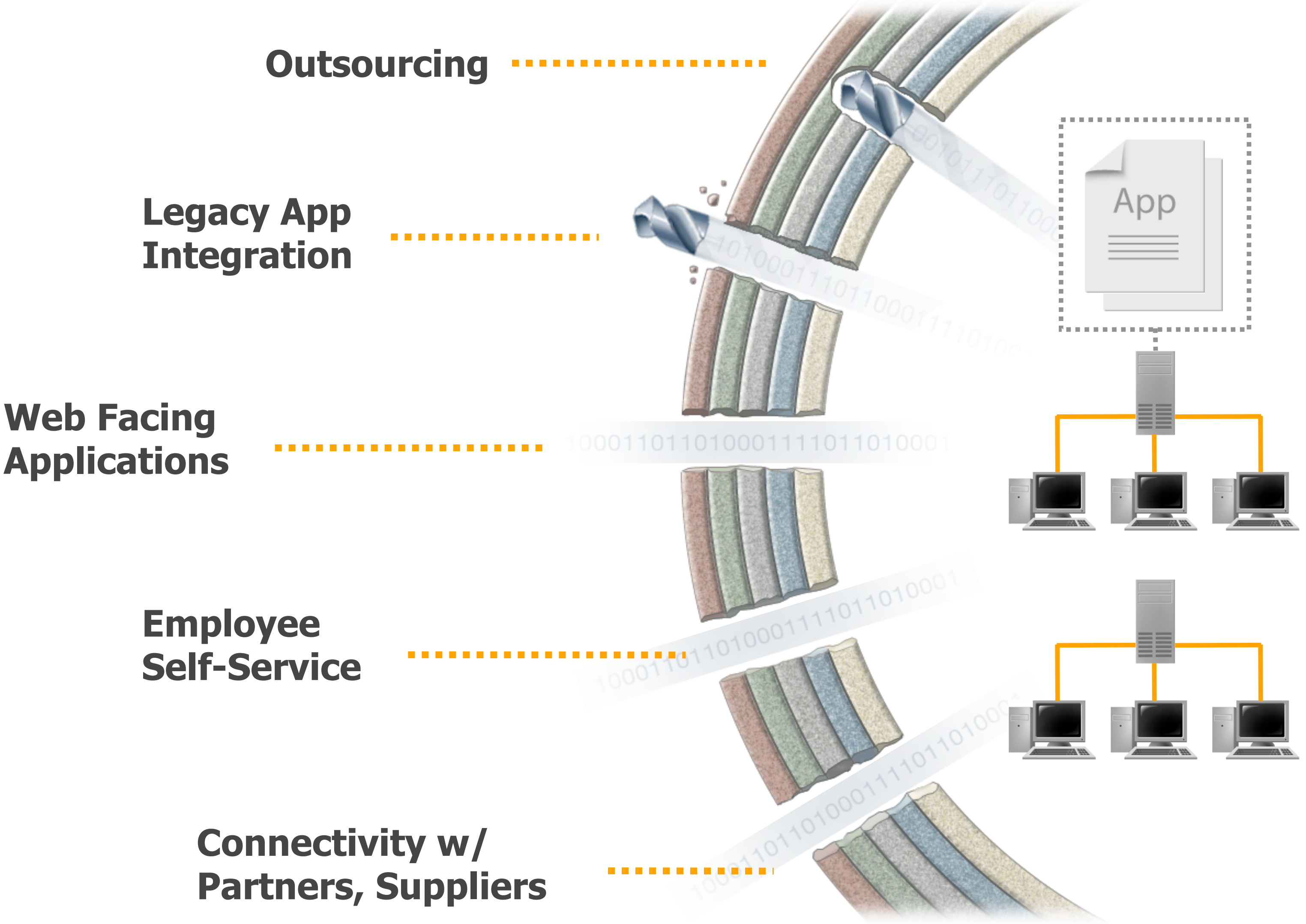
March 2007

CSIS

March 2007 Report

http://www.csis.org/media/csis/pubs/070323_lewisforeigninflubook.pdf

Applications Now Cut Through the Security Perimeter



Outsourcing

Legacy App Integration

Web Facing Applications

Employee Self-Service

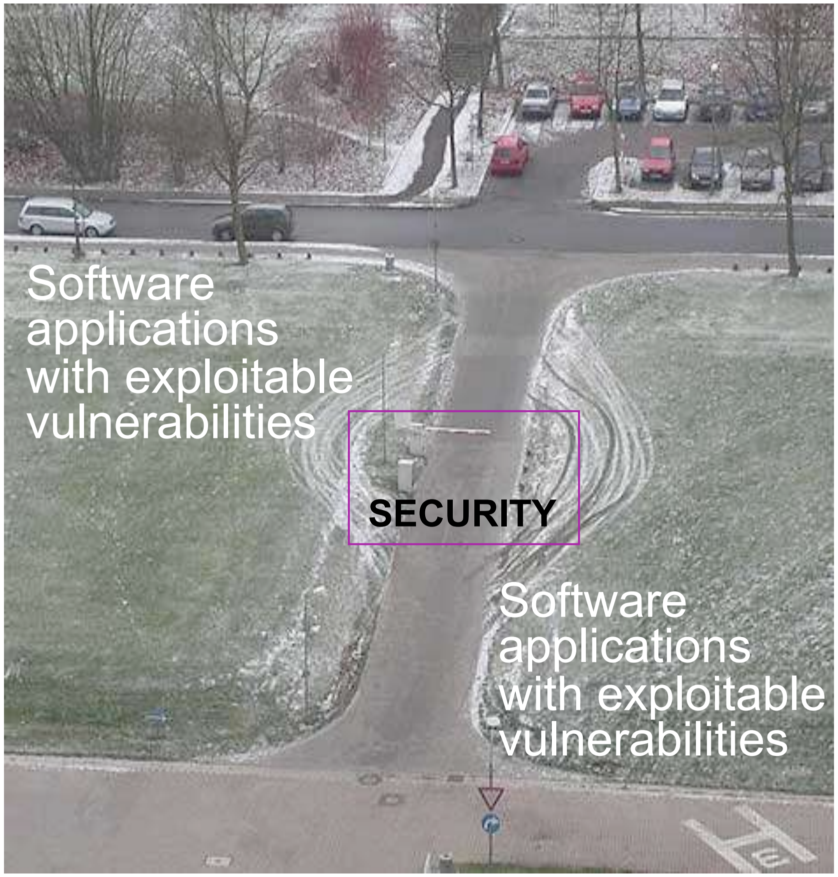
Connectivity w/ Partners, Suppliers

Homeland Security

“Neutralizing the Threat: A Case Study in Enterprise-wide Application Security Deployments,” Bruce C. Jenkins, Fortify Software

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

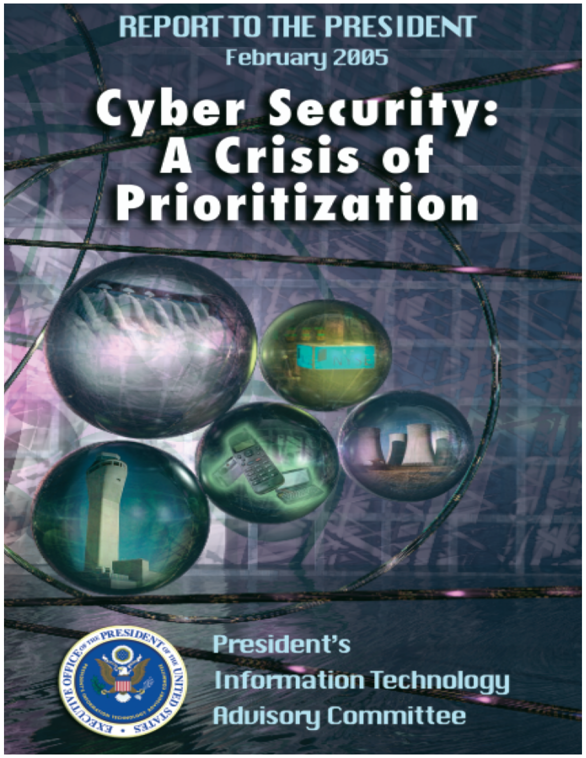

Software applications with exploitable vulnerabilities
SECURITY
Software applications with exploitable vulnerabilities

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

PITAC* Findings Relative to Needs for Secure Software Engineering & Software Assurance

- Commercial software engineering today lacks the scientific underpinnings and rigorous controls needed to produce high-quality, secure products at acceptable cost.
- Commonly used software engineering practices permit dangerous errors, such as improper handling of buffer overflows, which enable hundreds of attack programs to compromise millions of computers every year.
- In the future, the Nation may face even more challenging problems as adversaries – both foreign and domestic – become increasingly sophisticated in their ability to insert malicious code into critical software.
- **Recommendations for increasing investment in cyber security provided to NITRD Interagency Working Group for Cyber Security & Information Assurance R&D**



REPORT TO THE PRESIDENT
February 2005

Cyber Security:
A Crisis of
Prioritization

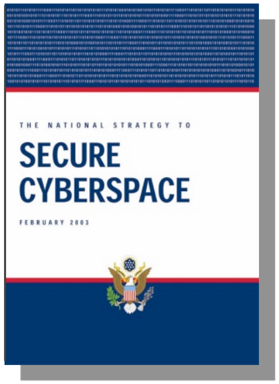
President's
Information Technology
Advisory Committee

* President's Information Technology Advisory Committee (PITAC) Report to the President, "Cyber Security: A Crisis of Prioritization," February 2005 identified top 10 areas in need of increased support, including: 'secure software engineering and software assurance' and 'metrics, benchmarks, and best practices' [Note: PITAC is now a part of PCAST]

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.

DHS Software Assurance Program Overview



THE NATIONAL STRATEGY TO

SECURE CYBERSPACE

FEBRUARY 2003

- 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."

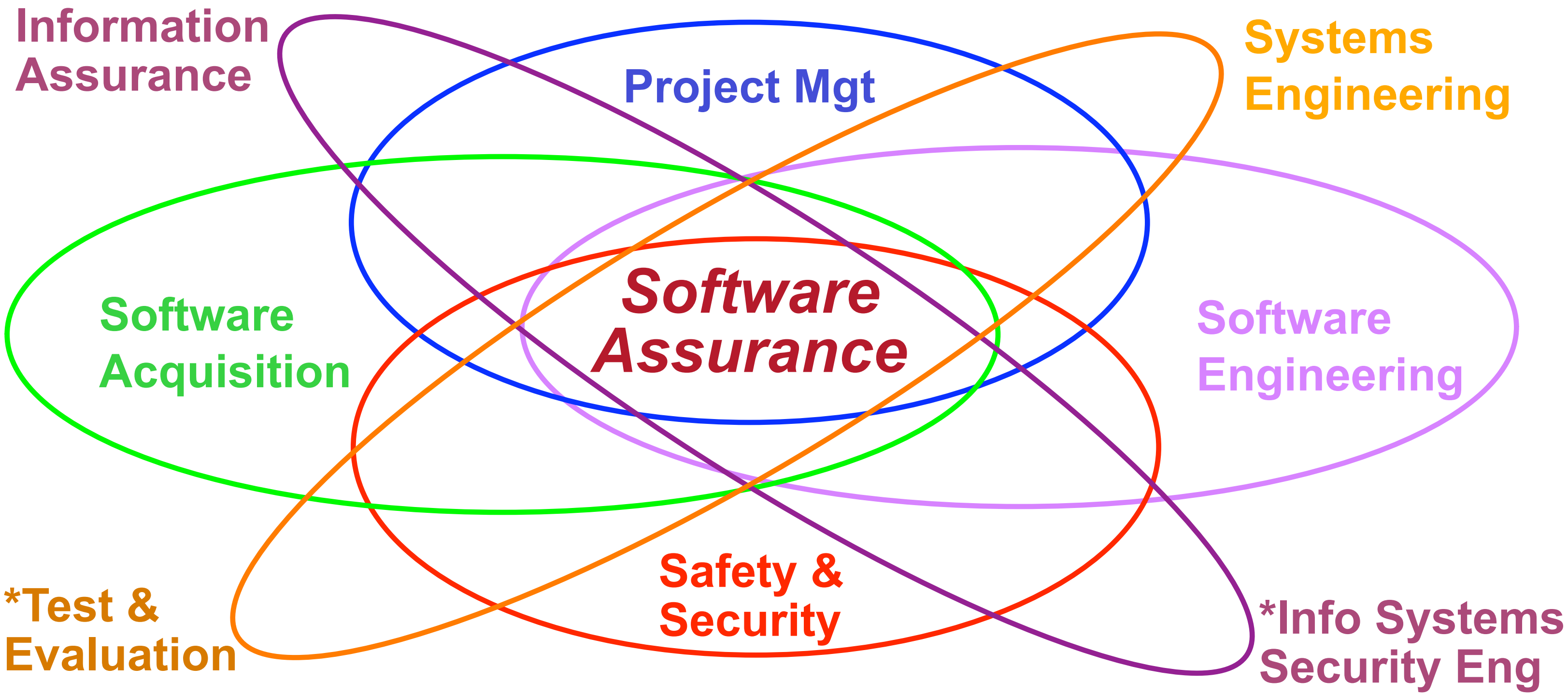
- DHS Program goals promote the **security and resilience** of software across the development, acquisition, and operational life cycle

- 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, and it will recover quickly to an acceptable level of operating capacity;
  - **Conformance** – Planned, systematic set of multi-disciplinary activities that ensure processes/products conform to requirements, standards/procedures.

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".

Homeland Security

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

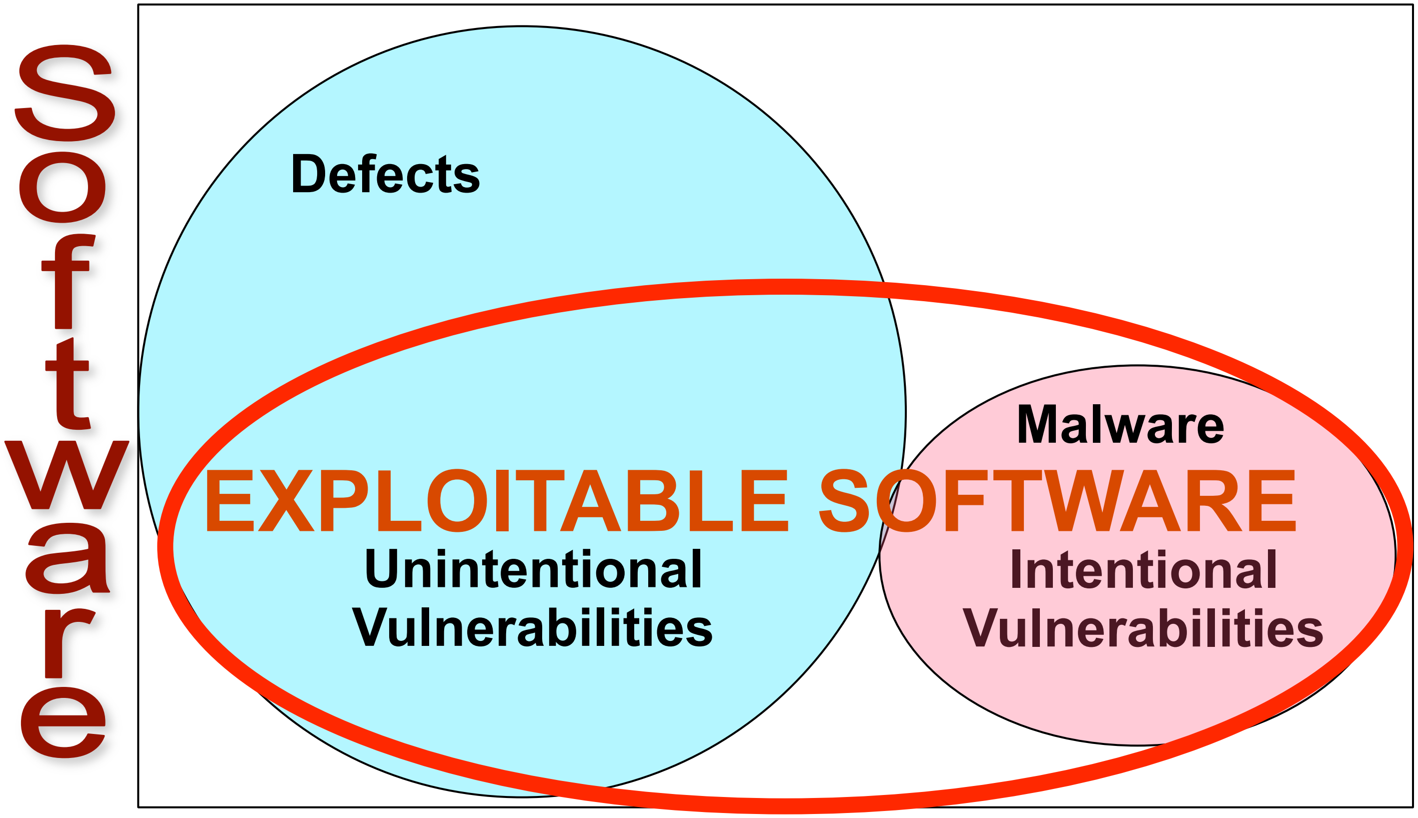
* See ‘Notes Page’ view for contributing BOK URLs and relevant links

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)

Homeland Security

Note: Chart is not to scale – notional representation -- for discussions

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

* 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*

BUILDING SECURITY IN
SOFTWARE ASSURANCE

… 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.

UK Secure Software Development Panel:

documenting key publications to define state of the art in 2009

1. Secure software development for human computer interaction;
2. Building and validating the behaviour and properties of software components;
3. Bench marking and best practice for secure software development;
4. Need to define academic standards/curriculum for teaching of secure software development;
5. How can we test large scale systems that required secure software development;
6. Development and analysis of business drivers to get suppliers to deliver secure software;
7. Development of source code analysis tools from research into insecure coding practices;
8. Understanding the economics of secure software development and the uptake of secure software development;
9. The measurement and analysis of trust and security as an emergent property in relation to secure software development;
10. How do we transfer research from secure software development into industry;
11. Understanding how we purchase and deliver secure software;
12. How do we accredit secure software;
13. How do we develop shared services and management the off-shoring software development process for secure software development;
14. How can OGS help in the procurement and development of secure software development?

SwA Collaboration for Content & Peer Review

Build Security In
Setting a higher standard for software assurance
Sponsored by DHS National Cyber Security Division

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

Software Assurance
Community Resources and Information Clearinghouse
Sponsored by DHS National Cyber Security Division

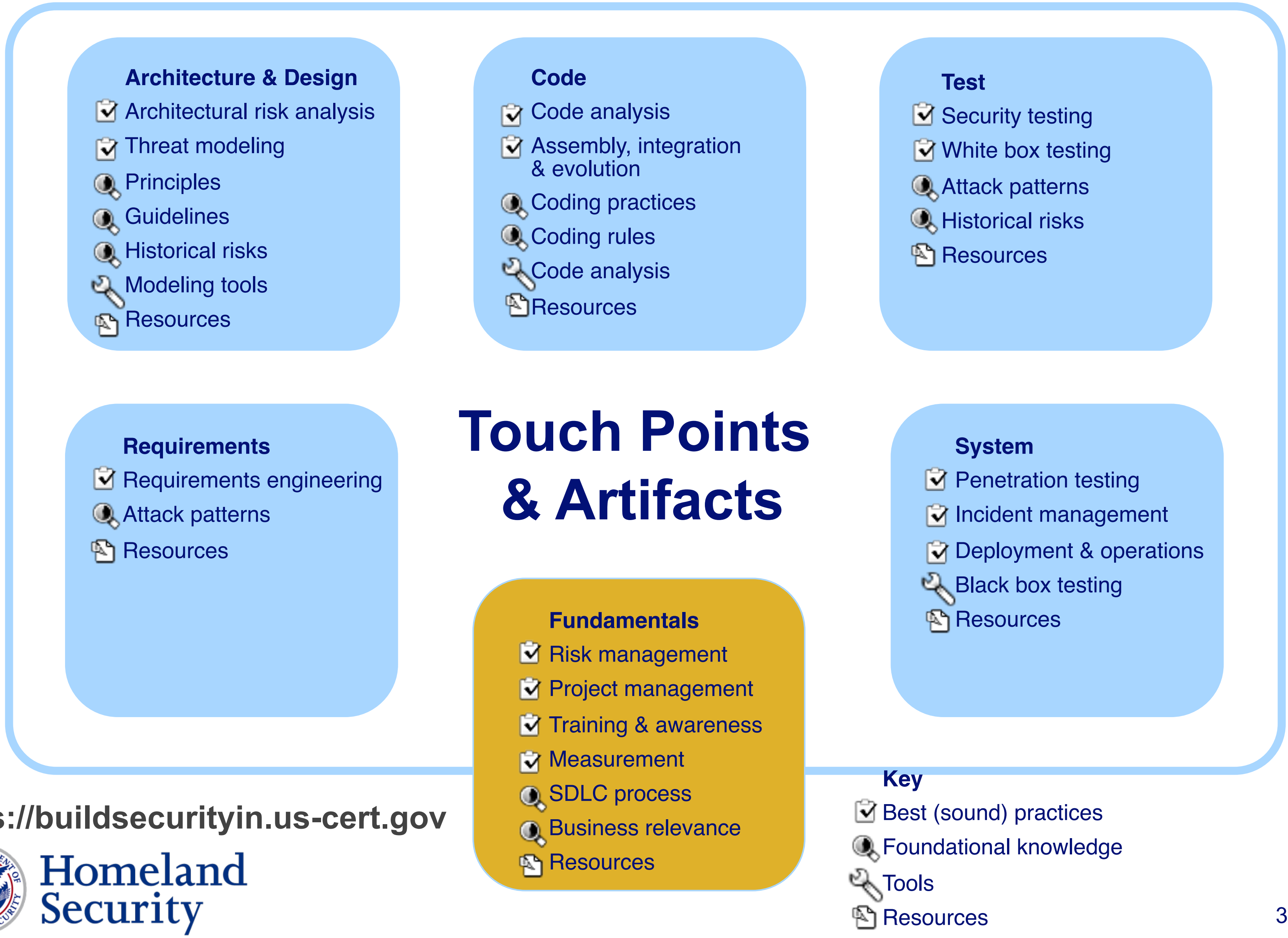
SwA Community Resources and Information Clearinghouse (CRIC)

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.

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

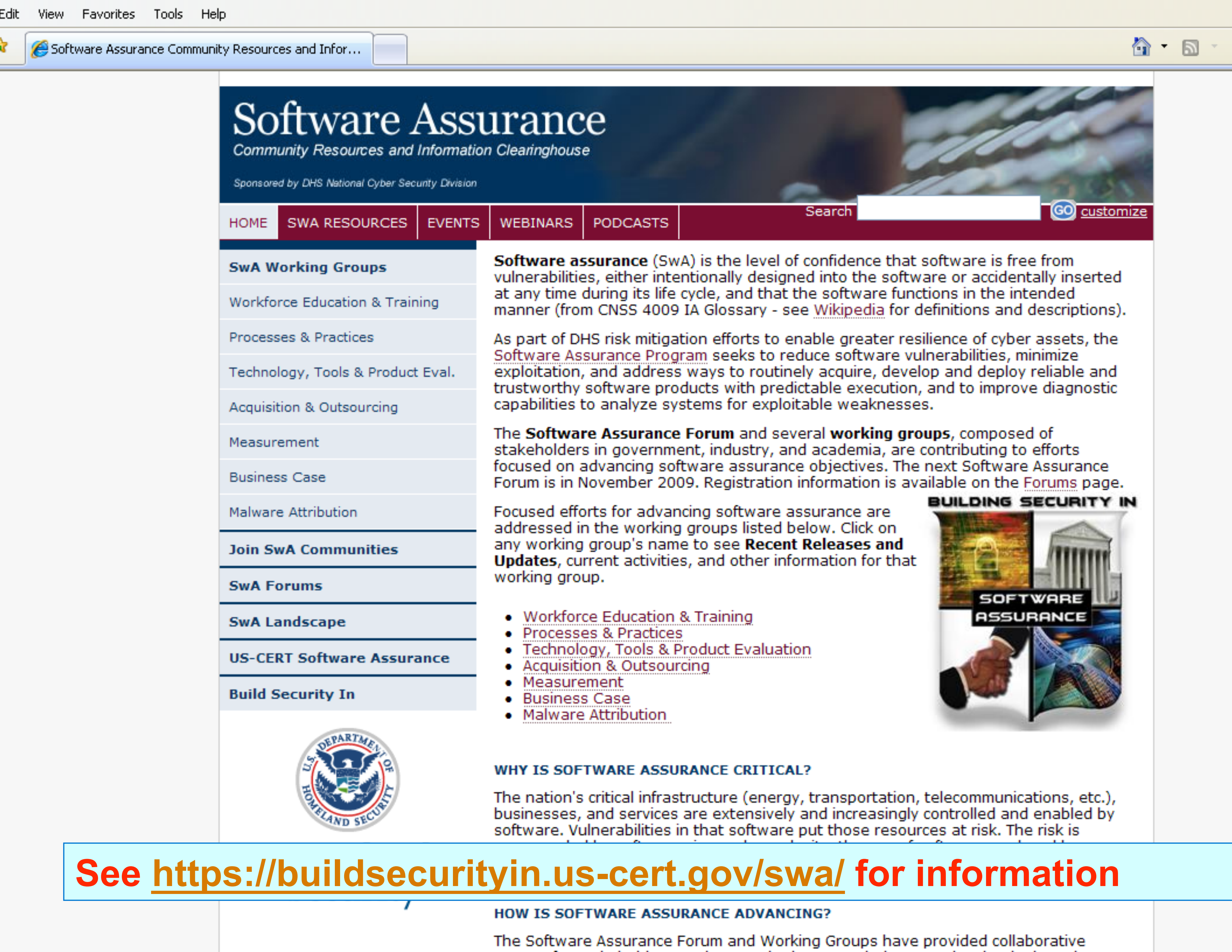
Requirements
- ☑ Requirements engineering
- 🔍 Attack patterns
- 📄 Resources

Touch Points & Artifacts

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

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

Key
- ☑ Best (sound) practices
- 🔍 Foundational knowledge
- 🔧 Tools
- 📄 Resources



Software Assurance

Community Resources and Information Clearinghouse

Sponsored by DHS National Cyber Security Division

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. The next Software Assurance Forum is in November 2009. Registration information is available on the Forums page.

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 Attribution

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

See https://buildsecurityin.us-cert.gov/swa/ for information

HOW IS SOFTWARE ASSURANCE ADVANCING?

The Software Assurance Forum and Working Groups have provided collaborative

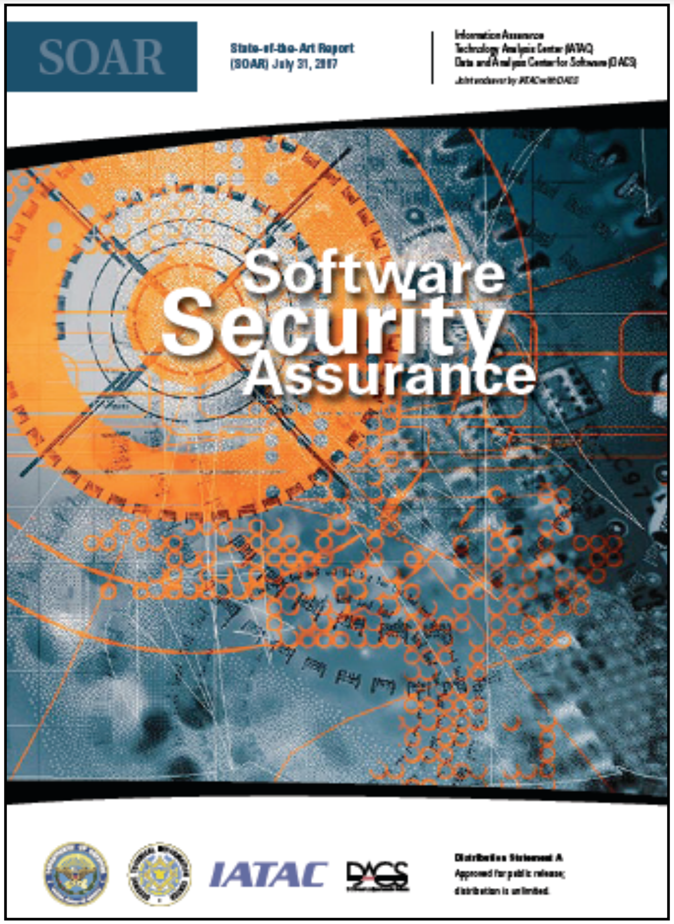
DHS Software Assurance (SwA) Outreach

- 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” to relevant stakeholders
  - 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”
- 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)

CROSSTALK
March 2007 The Journal of Defense Software Engineering Vol. 20 No. 3
SOFTWARE SECURITY
CROSSTALK
September 2006 The Journal of Defense Software Engineering Vol. 19 No. 9
SOFTWARE ASSURANCE
SPECIAL ISSUE: ENSURING SECURE
CROSSTAL
October 2005 The Journal of Defense Software Engineering
SOFTWARE SECURITY
Application Security

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



SOAR

State-of-the-Art Report
(SOAR) July 31, 2007

Information Assurance
Technology Analysis Center (IATAC)
Data and Analysis Center for Software (DACS)
Joint endeavor by IATAC with DACS

Software
Security
Assurance

IATAC DACS

Distribution Statement A
Approved for public release;
distribution is unlimited.

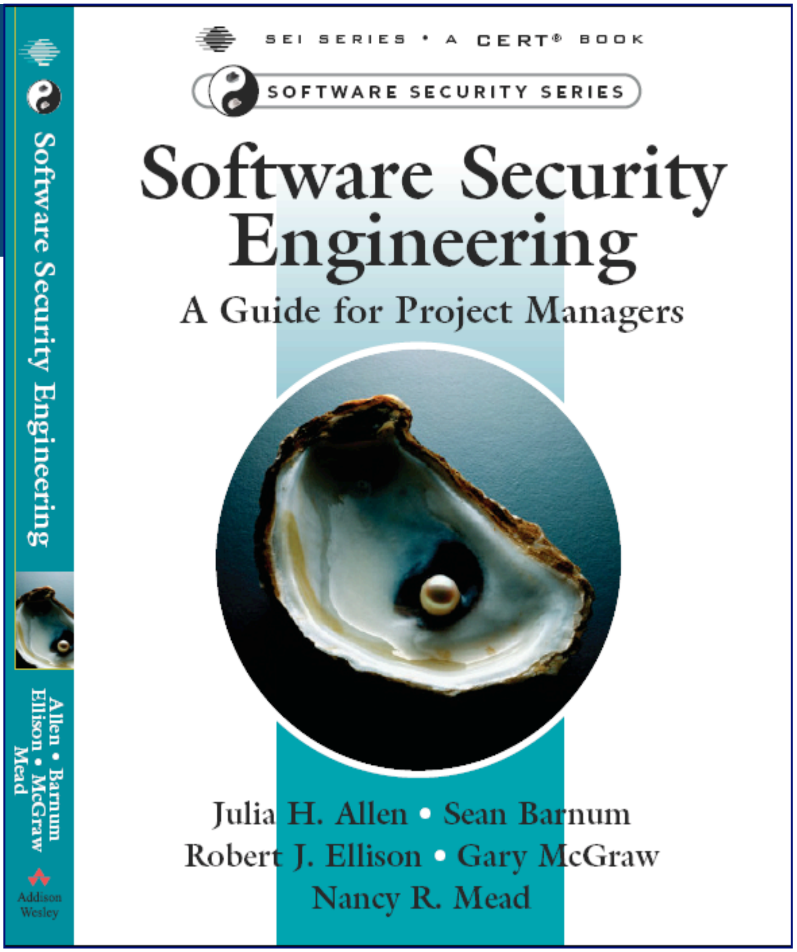
•The SOAR reflects output of efforts in the DoD-DHS Software Assurance Forum and Working Groups that provide collaborative venues for stakeholders to share and advance techniques and technologies relevant to software security.

Software Security Engineering:
A Guide for Project Managers

Build Security In

Setting a Higher Standard for Software Assurance

Sponsored by DHS National Cyber Security Division


SEI SERIES • A CERT® BOOK

SOFTWARE SECURITY SERIES

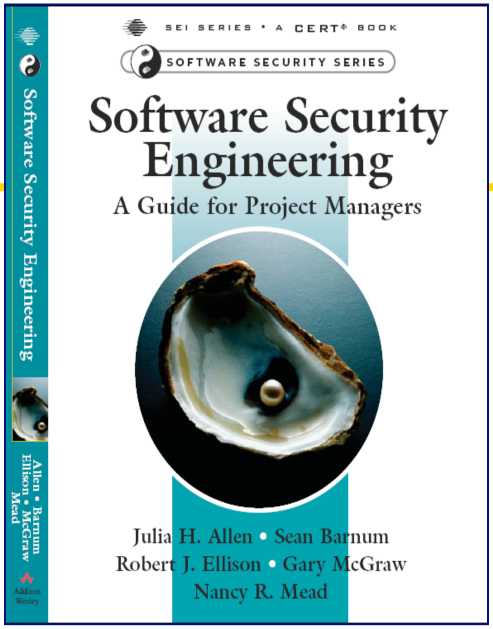
Software Security Engineering

A Guide for Project Managers

Julia H. Allen • Sean Barnum
Robert J. Ellison • Gary McGraw
Nancy R. Mead

- 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

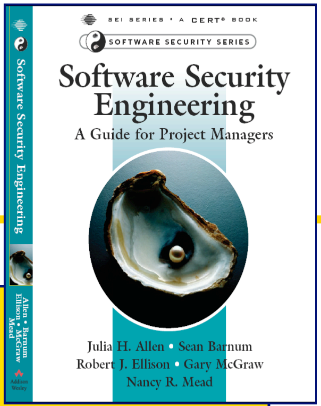
Software Security Engineering: A Guide for Project Managers



Six Main Practice Areas

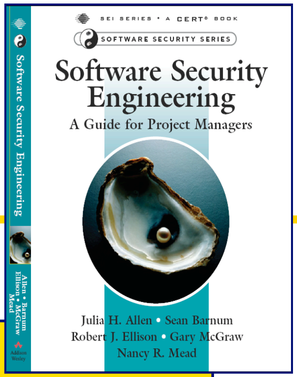
- Software security practices that span the SDLC
- Requirements engineering practices
- Architecture and design practices
- Coding and testing practices
- Security analysis for system complexity and scale: mitigations
- Governance and management practices

Software Security Engineering:
A Guide for Project Managers


SEI SERIES • A CERT® BOOK
SOFTWARE SECURITY SERIES
Software Security Engineering
A Guide for Project Managers
Julia H. Allen • Sean Barnum
Robert J. Ellison • Gary McGraw
Nancy R. Mead
Software Security Engineering
Allen • Barnum
Ellison • McGraw
Mead
Addison Wesley

Maturity Level	Description
L1	The content provides guidance for how to think about a topic for which there is no proven or widely accepted approach. The intent of the description is to raise awareness and aid the reader in thinking about the problem and candidate solutions. The content may also describe promising research results that may have been demonstrated in a constrained setting.
L2	The content describes practices that are in early pilot use and are demonstrating some successful results.
L3	The content describes practices that are in limited use in industry or government organizations, perhaps for a particular market sector.
L4	The content describes practices that have been successfully deployed and are in widespread use. Readers can start using these practices today with confidence. Experience reports and case studies are typically available.

Software Security Engineering:
A Guide for Project Managers


SEI SERIES • A CERT® BOOK
SOFTWARE SECURITY SERIES
Software Security Engineering
A Guide for Project Managers
Julia H. Allen • Sean Barnum
Robert J. Ellison • Gary McGraw
Nancy R. Mead
Addison Wesley

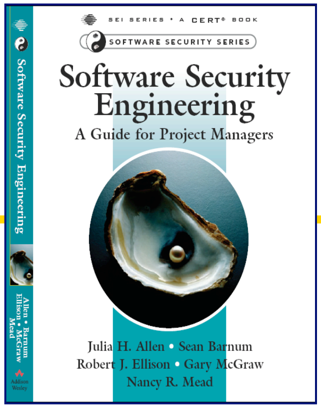
Audience Indicators

Audience Code	Description
E	executive and senior managers
M	project and mid-level managers
L	technical leaders, engineering managers, first line managers, and supervisors

Practices sorted and tagged as being relevant for respective roles:

- Executive responsible for software development
- Project manager
- Security analyst
- Requirements engineer
- Architect
- Designer
- Developer
- Quality assurance engineer
- Acquisition manager
- Software supplier
- All software engineering roles
- Stakeholders

Software Security Engineering:
A Guide for Project Managers



SEI SERIES • A CERT® BOOK

SOFTWARE SECURITY SERIES

Software Security Engineering

A Guide for Project Managers

Julia H. Allen • Sean Barnum
Robert J. Ellison • Gary McGraw
Nancy R. Mead

Software Security Engineering

Allen • Barnum
Ellison • McGraw
Mead

Addison Wesley

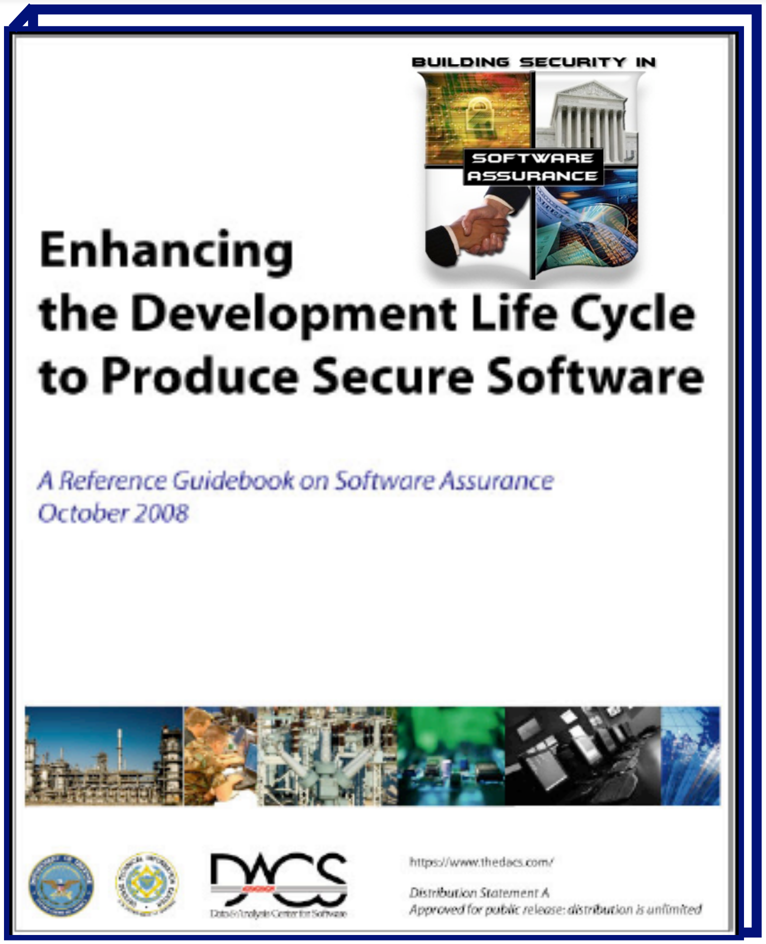
Recommendations

- Treat software security as a risk management issue
- Address software security in all contexts
  - Development, outsourcing, acquisition, purchase, with partners, hosting another party's product/service
- For internally developed software, integrate security practices into your SDLC
- Ensure applications have adequate controls for audit trails, and review these
- Tackle security as early in the life cycle as possible

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/



BUILDING SECURITY IN

SOFTWARE ASSURANCE

Enhancing the Development Life Cycle to Produce Secure Software

A Reference Guidebook on Software Assurance
October 2008

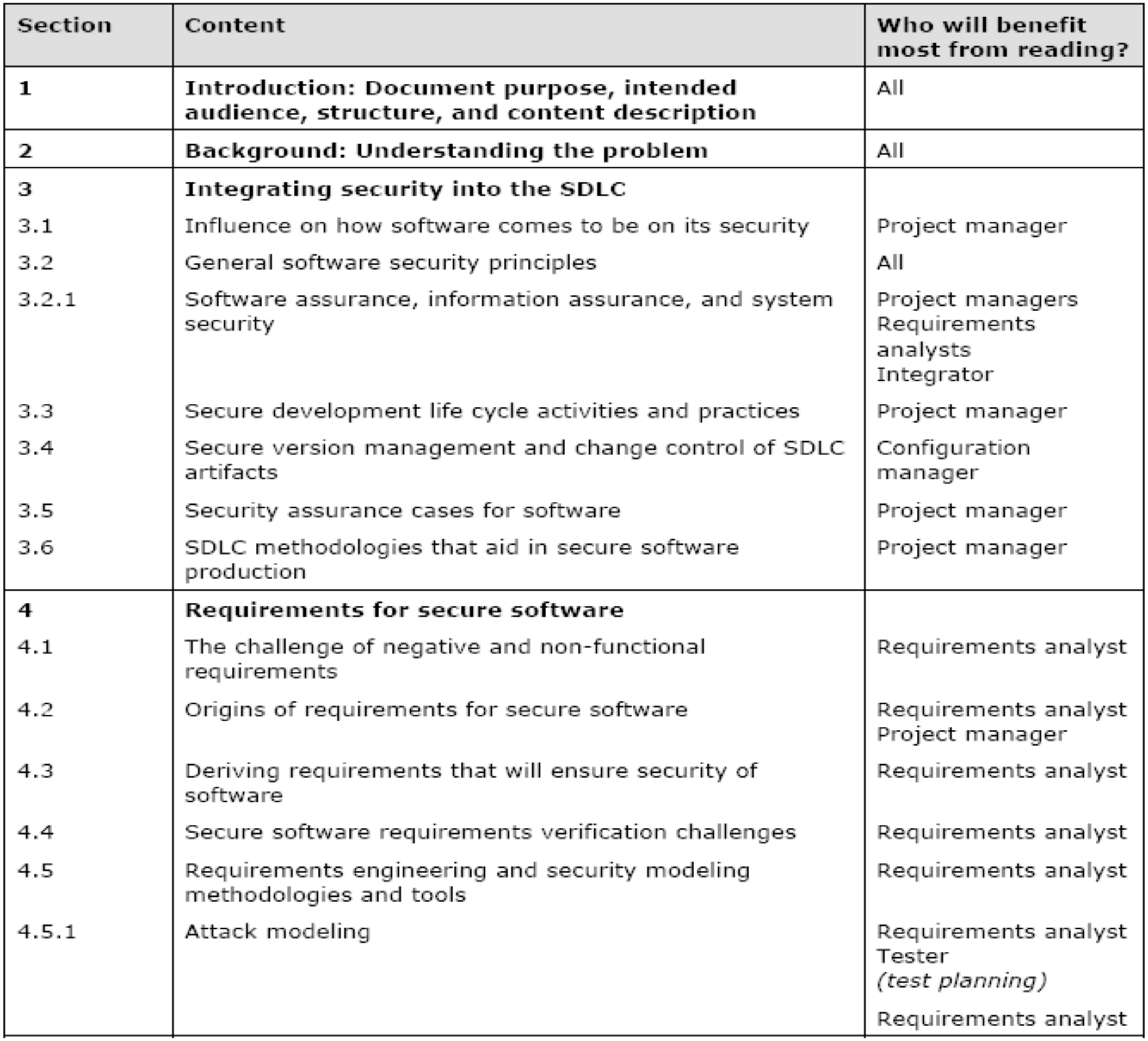
DACS
Data & Analysis Center for Software

https://www.thedacs.com/

Distribution Statement A
Approved for public release: distribution is unlimited

Enhancing the Development Life Cycle to Produce Secure Software

A Reference Guidebook on Software Assurance, October 2008



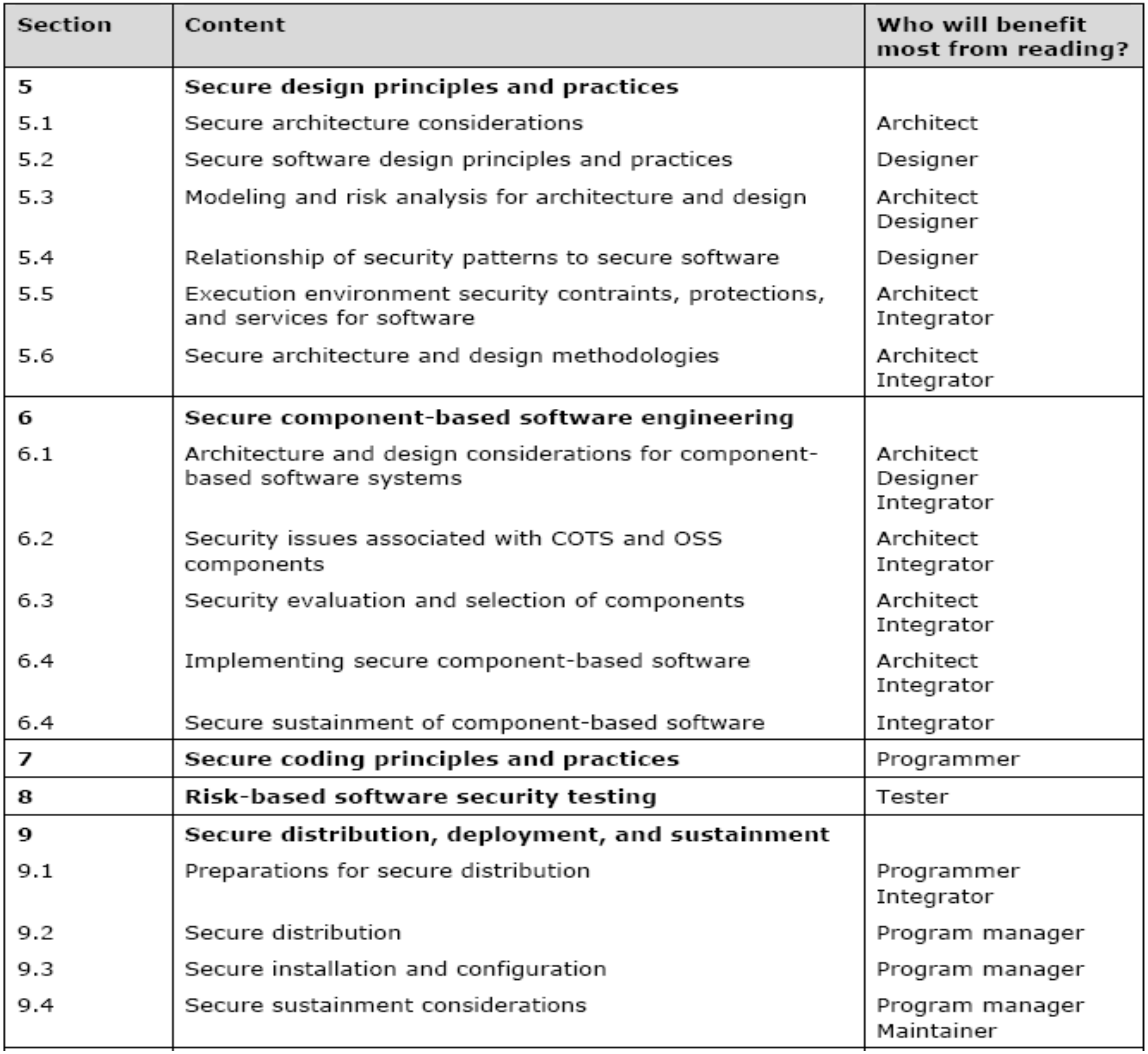
Section	Content	Who will benefit most from reading?
1	**Introduction: Document purpose, intended audience, structure, and content description**	All
2	**Background: Understanding the problem**	All
3	**Integrating security into the SDLC**	
3.1	Influence on how software comes to be on its security	Project manager
3.2	General software security principles	All
3.2.1	Software assurance, information assurance, and system security	Project managers Requirements analysts Integrator
3.3	Secure development life cycle activities and practices	Project manager
3.4	Secure version management and change control of SDLC artifacts	Configuration manager
3.5	Security assurance cases for software	Project manager
3.6	SDLC methodologies that aid in secure software production	Project manager
4	**Requirements for secure software**	
4.1	The challenge of negative and non-functional requirements	Requirements analyst
4.2	Origins of requirements for secure software	Requirements analyst Project manager
4.3	Deriving requirements that will ensure security of software	Requirements analyst
4.4	Secure software requirements verification challenges	Requirements analyst
4.5	Requirements engineering and security modeling methodologies and tools	Requirements analyst
4.5.1	Attack modeling	Requirements analyst Tester *(test planning)*
		Requirements analyst

BUILDING SECURITY IN

SOFTWARE ASSURANCE

Enhancing the Development Life Cycle to Produce Secure Software

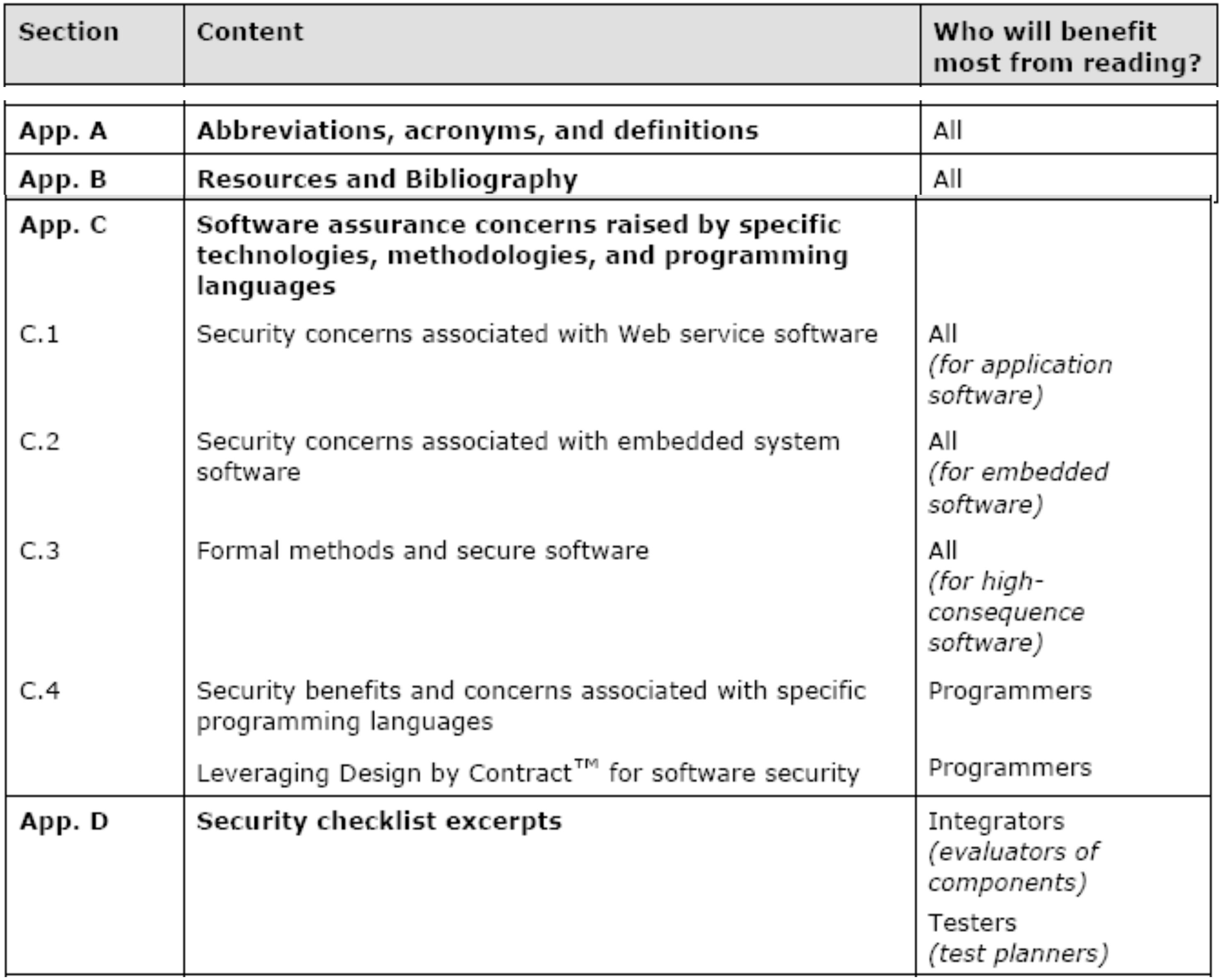
A Reference Guidebook on Software Assurance, October 2008



Section	Content	Who will benefit most from reading?
5	**Secure design principles and practices**	
5.1	Secure architecture considerations	Architect
5.2	Secure software design principles and practices	Designer
5.3	Modeling and risk analysis for architecture and design	Architect Designer
5.4	Relationship of security patterns to secure software	Designer
5.5	Execution environment security contraints, protections, and services for software	Architect Integrator
5.6	Secure architecture and design methodologies	Architect Integrator
6	**Secure component-based software engineering**	
6.1	Architecture and design considerations for component-based software systems	Architect Designer Integrator
6.2	Security issues associated with COTS and OSS components	Architect Integrator
6.3	Security evaluation and selection of components	Architect Integrator
6.4	Implementing secure component-based software	Architect Integrator
6.4	Secure sustainment of component-based software	Integrator
7	**Secure coding principles and practices**	Programmer
8	**Risk-based software security testing**	Tester
9	**Secure distribution, deployment, and sustainment**	
9.1	Preparations for secure distribution	Programmer Integrator
9.2	Secure distribution	Program manager
9.3	Secure installation and configuration	Program manager
9.4	Secure sustainment considerations	Program manager Maintainer

Enhancing the Development Life Cycle to Produce Secure Software

A Reference Guidebook on Software Assurance, October 2008



Section	Content	Who will benefit most from reading?
App. A	**Abbreviations, acronyms, and definitions**	All
App. B	**Resources and Bibliography**	All
App. C	**Software assurance concerns raised by specific technologies, methodologies, and programming languages**	
C.1	Security concerns associated with Web service software	All *(for application software)*
C.2	Security concerns associated with embedded system software	All *(for embedded software)*
C.3	Formal methods and secure software	All *(for high-consequence software)*
C.4	Security benefits and concerns associated with specific programming languages	Programmers
	Leveraging Design by Contract™ for software security	Programmers
App. D	**Security checklist excerpts**	Integrators *(evaluators of components)* Testers *(test planners)*

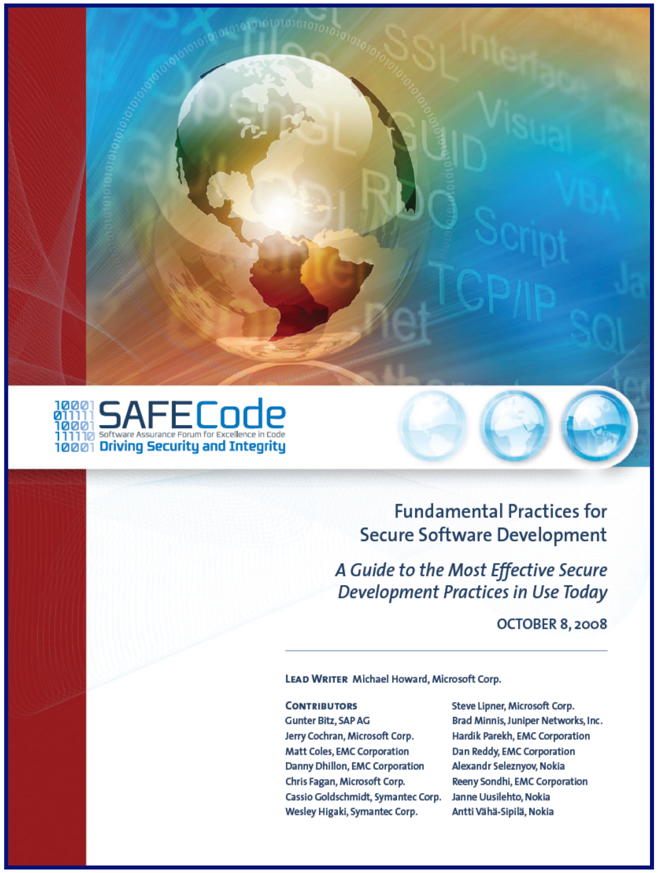
BUILDING SECURITY IN

SOFTWARE ASSURANCE

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)



SAFECode
Software Assurance Forum for Excellence in Code
Driving Security and Integrity

Fundamental Practices for
Secure Software Development

*A Guide to the Most Effective Secure
Development Practices in Use Today*

OCTOBER 8, 2008

LEAD WRITER Michael Howard, Microsoft Corp.

CONTRIBUTORS
Gunter Bitz, SAP AG
Jerry Cochran, Microsoft Corp.
Matt Coles, EMC Corporation
Danny Dhillon, EMC Corporation
Chris Fagan, Microsoft Corp.
Cassio Goldschmidt, Symantec Corp.
Wesley Higaki, Symantec Corp.
Steve Lipner, Microsoft Corp.
Brad Minnis, Juniper Networks, Inc.
Hardik Parekh, EMC Corporation
Dan Reddy, EMC Corporation
Alexandr Seleznyov, Nokia
Reeny Sondhi, EMC Corporation
Janne Uusilehto, Nokia
Antti Vähä-Sipilä, Nokia

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

Software Assurance:
An Overview of Current Industry Best Practices, February 2008

- The Challenge of Software Assurance and Security
- Industry Best Practices for Software Assurance and Security
- Framework for Software Development
- Software Security Best Practices
- Related Roles of Integrators and End Users
- SAFECode's Goals
- Questions for Vendors about Product Assurance and Security
- About SAFECode

SAFECode

Software Assurance Forum for Excellence in Code

Driving Security and Integrity

Software Assurance:
An Overview of
Current Industry Best Practices

February 2008

http://www.safecode.org/publications/SAFECode_BestPractices0208.pdf

SOFTWARE ASSURANCE FORUM

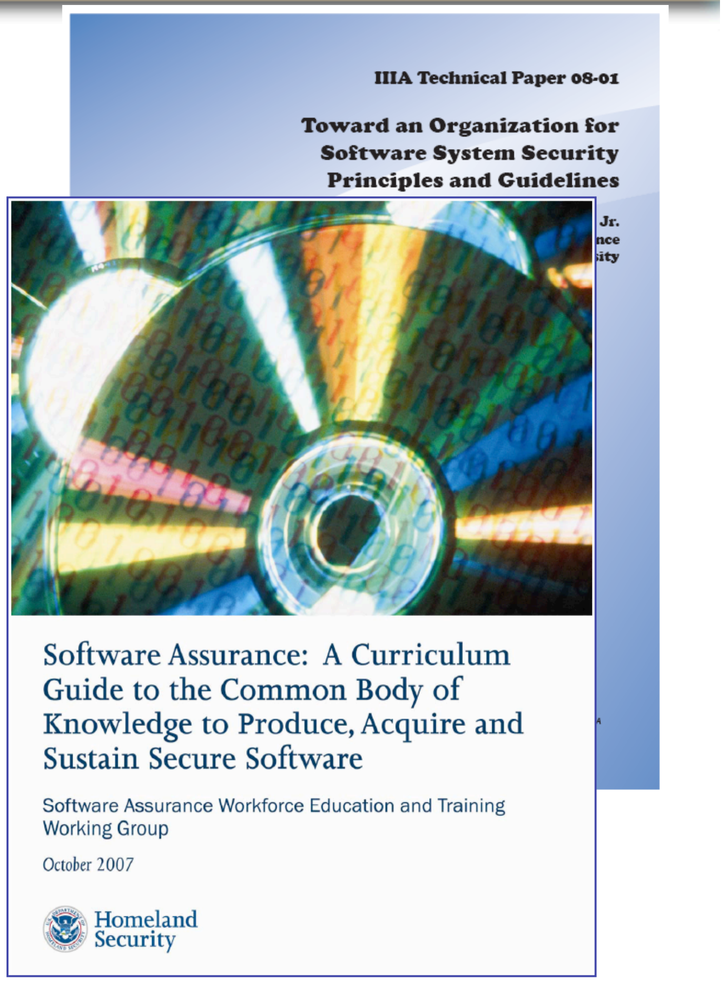
BUILDING SECURITY IN

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



IIIA Technical Paper 08-01

Toward an Organization for Software System Security Principles and Guidelines

Jr.
nce
ity

Software Assurance: A Curriculum Guide to the Common Body of Knowledge to Produce, Acquire and Sustain Secure Software

Software Assurance Workforce Education and Training Working Group

October 2007

Homeland Security

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

Structuring Software Assurance CBK Content for Curricula Considerations

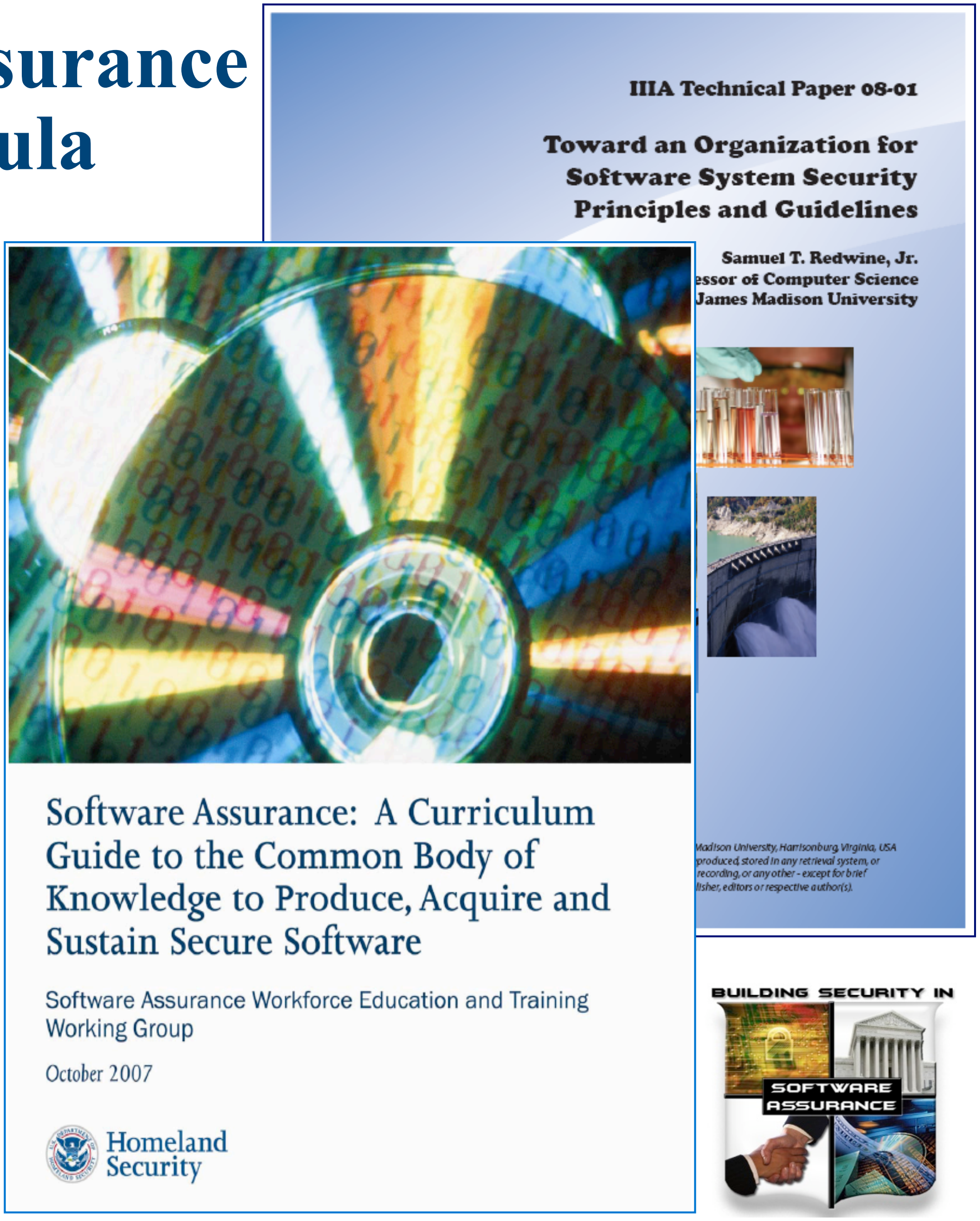
"Toward an Organization for Software System Security Principles and Guidelines," Version 1.0, IIIA Technical Paper 08-01. Feb 2008

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

Both collaboratively developed through the Software Assurance Working Group on Workforce Education and Training Co-chair Samuel T. Redwine, Jr.,

Institute for Infrastructure and Information Assurance,

James Madison University



IIIA Technical Paper 08-01

Toward an Organization for Software System Security Principles and Guidelines

Samuel T. Redwine, Jr.
essor of Computer Science
James Madison University

Madison University, Harrisonburg, Virginia, USA
produced, stored in any retrieval system, or
recording, or any other - except for brief
lisher, editors or respective author(s).

Software Assurance: A Curriculum Guide to the Common Body of Knowledge to Produce, Acquire and Sustain Secure Software

Software Assurance Workforce Education and Training Working Group

October 2007

Homeland Security

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

Toward an Organization for Software System Security Principles and Guidelines

0. INTRODUCTION

0.1/0.2 PURPOSE / SCOPE

0.3 REASONING UNDERLYING THE ORGANIZATION

0.4 ORGANIZATION OF REMAINDER OF DOCUMENT

1. THE ADVERSE

1.1. LIMIT, REDUCE, OR MANAGE VIOLATORS

1.2. LIMIT, REDUCE, OR MANAGE BENEFITS TO VIOLATORS OR ATTACKERS

1.3. INCREASE ATTACKER LOSSES

1.4. INCREASE ATTACKER UNCERTAINTY

2. THE SYSTEM

2.1. LIMIT, REDUCE, OR MANAGE VIOLATIONS

2.2. IMPROVE BENEFITS OR AVOID ADVERSE EFFECTS ON SYSTEM BENEFITS

2.3. LIMIT, REDUCE, OR MANAGE SECURITY-RELATED COSTS

2.4. LIMIT, REDUCE, OR MANAGE SECURITY-RELATED UNCERTAINTIES

3. THE ENVIRONMENT

3.1. NATURE OF ENVIRONMENT

3.2. BENEFITS TO AND FROM ENVIRONMENT

3.3. LIMIT, REDUCE, OR MANAGE ENVIRONMENT-RELATED LOSSES

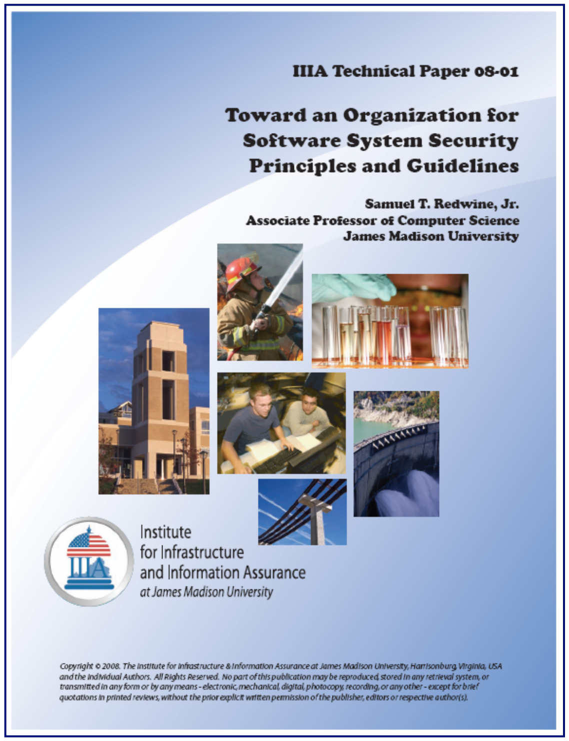
3.4. LIMIT, REDUCE, OR MANAGE ENVIRONMENT-RELATED UNCERTAINTIES

4. CONCLUSION

5. APPENDIX A: PRINCIPLES OF WAR

6. APPENDIX B: PURPOSE-CONDITION-ACTION-RESULT MATRIX

7/8. BIBLIOGRAPHY / ACKNOWLEDGEMENTS



IIIA Technical Paper 08-01

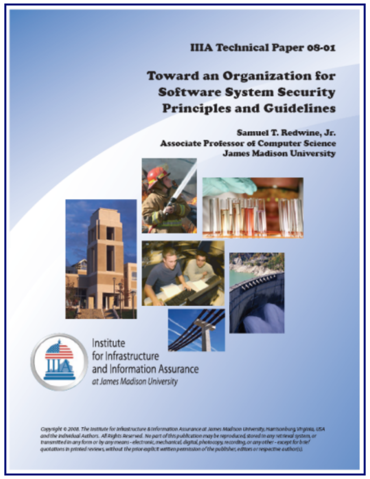
Toward an Organization for Software System Security Principles and Guidelines

Samuel T. Redwine, Jr.
Associate Professor of Computer Science
James Madison University

Institute
for Infrastructure
and Information Assurance
at James Madison University

BUILDING SECURITY IN
SOFTWARE ASSURANCE

Toward an Organization for Software System Security Principles and Guidelines



IIIA Technical Paper 08-01

Toward an Organization for Software System Security Principles and Guidelines

Samuel T. Redwine, Jr.
Associate Professor of Computer Science
James Madison University

Institute
for Infrastructure
and Information Assurance
at James Madison University

1. THE ADVERSE

1.1. LIMIT, REDUCE, OR MANAGE VIOLATORS

- *1.1.1. Adversaries are Intelligent and Malicious*
- *1.1.2. Limit, Reduce, or Manage Set of Violators*
- *1.1.3. Limit, Reduce, or Manage Attempted Violations*
- *1.1.4. Think like an Attacker*

1.2. LIMIT, REDUCE, OR MANAGE BENEFITS TO VIOLATORS OR ATTACKERS

- *1.2.1. Unequal Attacker Benefits and Defender Losses*
- *1.2.2. Limit, Reduce, or Manage Violators' Ability to Exploit Success for Gain*

1.3. INCREASE ATTACKER LOSSES

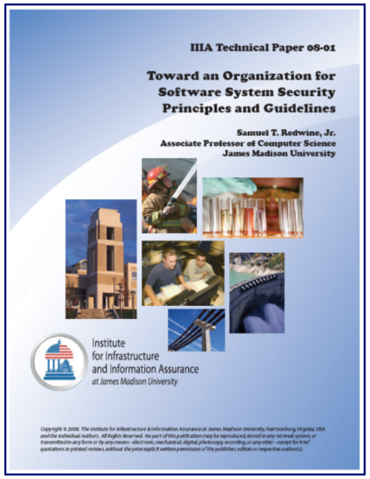
- *1.3.1. Limit, Reduce, Manage Violators' Ease in Taking Steps towards Violation*
- *1.3.2. Increase Losses and Likely Penalties for Preparation*
- *1.3.3. Increase Expense of Attacking*
- *1.3.4. Increase Attacker Losses and Likely Penalties*

1.4. INCREASE ATTACKER UNCERTAINTY

- *1.4.1. Conceal Information Useful to Attacker*
- *1.4.2. Exploit Deception*

BUILDING SECURITY IN
SOFTWARE ASSURANCE

Toward an Organization for Software System Security Principles and Guidelines



IIIA Technical Paper 08-01

Toward an Organization for Software System Security Principles and Guidelines

Samuel T. Redwine, Jr.
Associate Professor of Computer Science
James Madison University

Institute
for Infrastructure
and Information Assurance
at James Madison University

2. THE SYSTEM

2.1. LIMIT, REDUCE, OR MANAGE VIOLATIONS

- *2.1.1. Specify Security Requirements*
- *2.1.2. Limit, Reduce, or Manage Opportunities for Violations*
- *2.1.3. Limit Reduce, or Manage Actual Violations*
- *2.1.4. Limit, Reduce, or Manage Lack of Accountability*

2.2. IMPROVE BENEFITS OR AVOID ADVERSE EFFECTS ON SYSTEM BENEFITS

- *2.2.1. Access Fulfills Needs and Facilitates User*
- *2.2.2. Encourage and Ease Use of Security Aspects*
- *2.2.3. Articulate the Desired Characteristics and Tradeoff among Them*
- *2.2.4. Efficient Security*
- *2.2.5. Provide Added Benefits*
- *2.2.6. Learn, Adapt, and Improve*

2.3. LIMIT, REDUCE, OR MANAGE SECURITY-RELATED COSTS

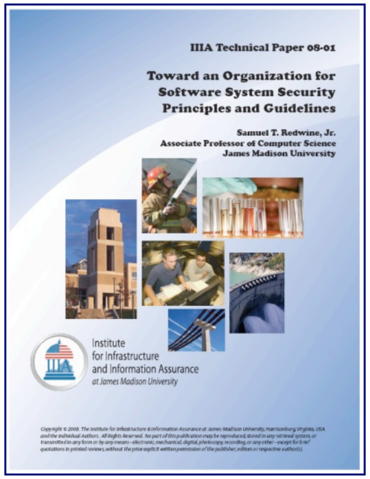
- *2.3.1. Limit, Reduce, or Manage Security-Related Adverse Consequences*
- *2.3.2. Limit, Reduce, or Manage Security-Related Expenses across the Lifecycle*

2.4. LIMIT, REDUCE, OR MANAGE SECURITY-RELATED UNCERTAINTIES

- *2.4.1. Identify Uncertainties*
- *2.4.2. Limit, Reduce, or Manage Security-Related Unknowns*
- *2.4.3. Limit, Reduce, or Manage Security-Related Assumptions*
- *2.4.4. Limit, Reduce, or Manage Lack of Integrity or Validity*
- *2.4.5. Limit, Reduce, or Manage Lack of Reliability or Availability of Security-related Resources*
- *2.4.6. Predictability – Limit, Reduce, or Manage Unpredictability of System Behavior*
- *2.4.7. Informed Consent*
- *2.4.8. Limit, Reduce, or Manage Consequences or Risks related to Uncertainty*
- *2.4.9. Increase Assurance regarding Product*

BUILDING SECURITY IN
SOFTWARE ASSURANCE

Toward an Organization for Software System Security Principles and Guidelines



IIIA Technical Paper 08-01

Toward an Organization for Software System Security Principles and Guidelines

Samuel T. Redwine, Jr.
Associate Professor of Computer Science
James Madison University

Institute
for Infrastructure
and Information Assurance
at James Madison University

3. THE ENVIRONMENT

3.1. NATURE OF ENVIRONMENT

3.1.1. Security is a System, Organizational, and Societal Problem
3.1.2. The Conflict Extents beyond Computing
3.1.3. New Technologies Have Security Problems

3.2. BENEFITS TO AND FROM ENVIRONMENT

3.2.1. Utilize Security Mechanisms Existing in Environment to Enhance One's Security
3.2.2. Create, Learn, and Adapt and Improve Organizational Policy
3.2.3. Learn from Environment
3.2.4. Help, but do not Help Attackers

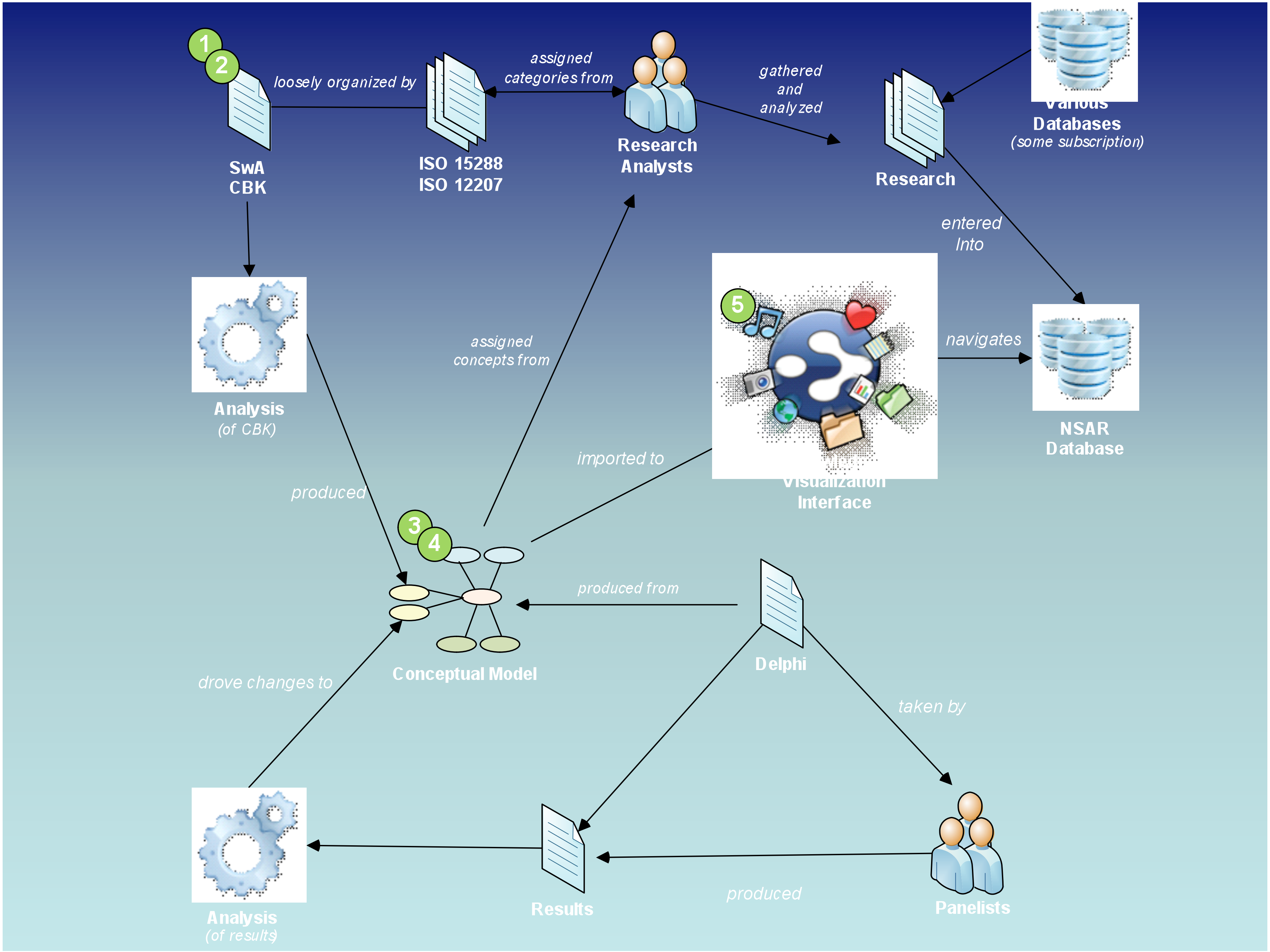
BUILDING SECURITY IN
SOFTWARE ASSURANCE

3.3. LIMIT, REDUCE, OR MANAGE ENVIRONMENT-RELATED LOSSES

3.3.1. Do Not Cause Security Problems for Systems in the Environment
3.3.2. Do Not Thwart Security Mechanisms in Environment
3.3.3. Avoid Dependence
3.3.4. Presume Environment is Dangerous

3.4. LIMIT, REDUCE, OR MANAGE ENVIRONMENT-RELATED UNCERTAINTIES

3.4.1. Know One's Environment
3.4.2. Limit, Reduce, or Manage Trust
3.4.3. Ensure Adequate Assurance for Dependences
3.4.4. Third-Parties are Sources of Uncertainty


1
2
SwA CBK
loosely organized by
ISO 15288
ISO 12207
assigned categories from
Research Analysts
gathered and analyzed
Research
Various Databases
(some subscription)
entered Into
Analysis
(of CBK)
5
Visualization Interface
navigates
NSAR Database
assigned concepts from
imported to
produced
3
4
Conceptual Model
produced from
Delphi
drove changes to
taken by
Analysis
(of results)
Results
produced
Panelists

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

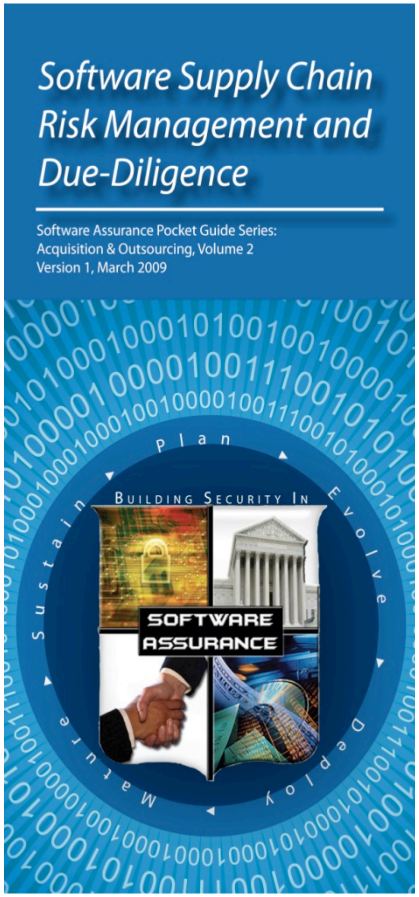
- Integrating Security into the Software Development Life Cycle
- Key Practices for Mitigating the Most Egregious Exploitable Software Weaknesses
- Risk-based Software Security Testing
- Requirements and Analysis for Secure Software
- Architecture and Design Considerations for Secure Software
- Secure 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



Software Supply Chain Risk Management and Due-Diligence

Software Assurance Pocket Guide Series:
Acquisition & Outsourcing, Volume 2
Version 1, March 2009

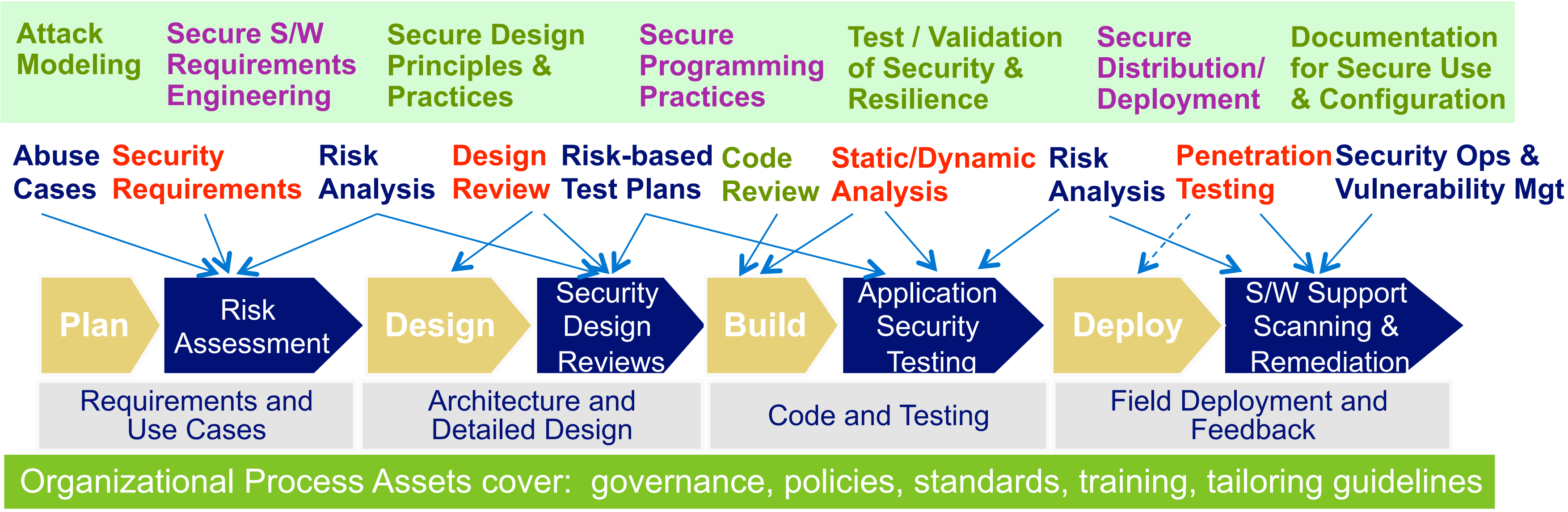
SwA Pocket Guides and SwA-related documents are collaboratively developed with peer review; they are subject to update and are freely available for download via the DHS Software Assurance Community Resources and Information Clearinghouse at https://buildsecurityin.us-cert.gov/swa (see SwA Resources)

BUILDING SECURITY IN
SOFTWARE ASSURANCE

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

Abuse Cases | Security Requirements | Risk Analysis | Design Review | Risk-based Test Plans | Code Review | Static/Dynamic Analysis | Risk Analysis | Penetration Testing | Security Ops & Vulnerability Mgt

Plan → Risk Assessment → Design → Security Design Reviews → Build → Application Security Testing → Deploy → S/W Support Scanning & Remediation

Requirements and Use Cases | Architecture and Detailed Design | Code and Testing | Field Deployment and Feedback

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

12. 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

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].

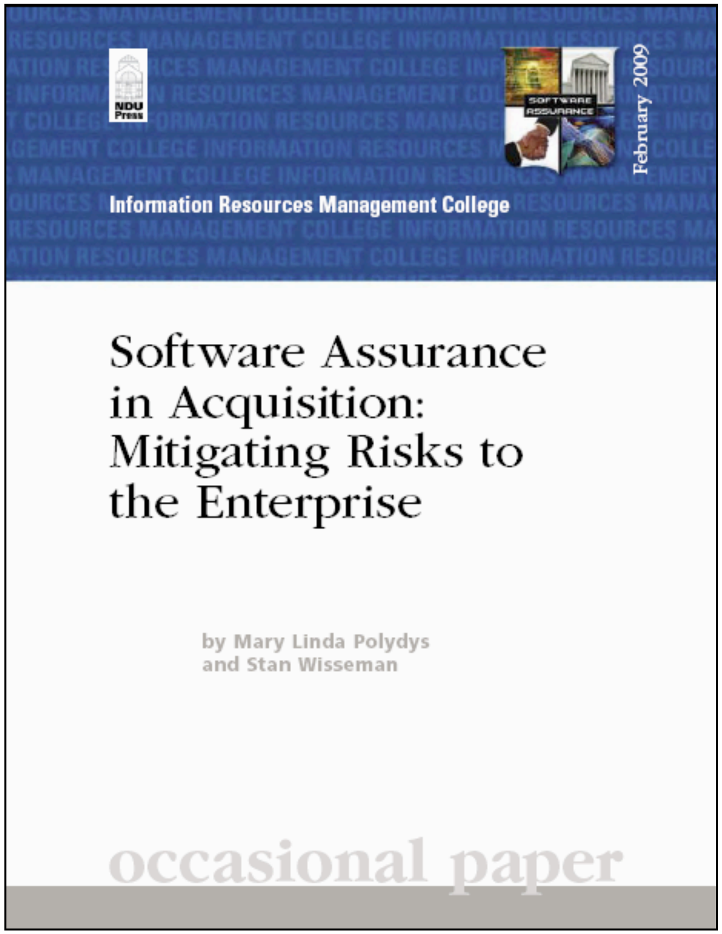
SwA Acquisition & Outsourcing Handbook

"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



NDU Press

February 2009

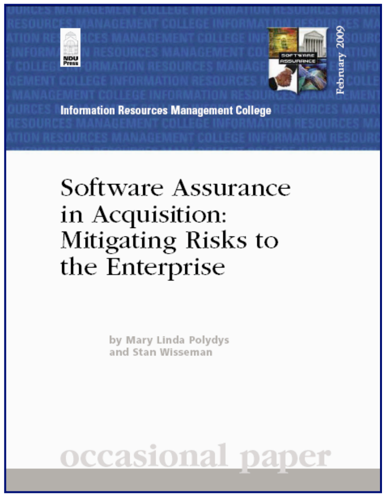
Information Resources Management College

Software Assurance in Acquisition: Mitigating Risks to the Enterprise

by Mary Linda Polydys
and Stan Wisseman

occasional paper

SwA Acquisition & Outsourcing Handbook



NDU Press

February 2009

Information Resources Management College

Software Assurance in Acquisition: Mitigating Risks to the Enterprise

by Mary Linda Polydys and Stan Wisseman

occasional paper

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 Propri-etary	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)

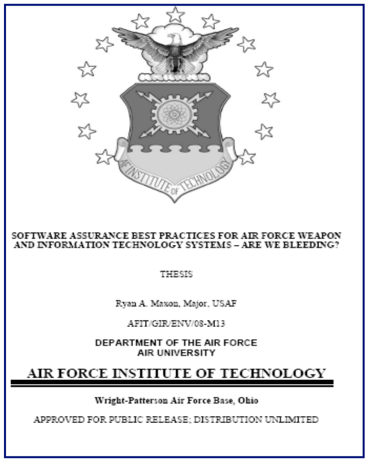
SwA Concern Categories	Risks	Purpose for Questions
Service Confidentiality Policies	Without policies to enforce client data confidentiality/ privacy, acquirer's data could be at risk without service supplier liability.	To determine the service provider's confidentiality and privacy policies and ensure 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?

Software Assurance Best Practices for Air Force Weapon and Information Technology Systems – Are We Bleeding?

AFIT Masters Thesis, March 2008, Major Ryan Maxon



SOFTWARE ASSURANCE BEST PRACTICES FOR AIR FORCE WEAPON AND INFORMATION TECHNOLOGY SYSTEMS – ARE WE BLEEDING?

THESIS

Ryan A. Maxon, Major, USAF

AFIT/GIR/ENV/08-M13

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

Sample of recommendations that should be implemented, including:

- Focus software-related practices on Four P‘s:
  - (1) —Practices for creating and updating software in a software assurance environment,
  - (2) —Processes supporting software assurance practices,
  - (3) —Protection from threats to code during and after development, and
  - (4) —Pedigree of those involved in software development/ follow-on process
- Provide Request for Proposal (RFP) and Statement of Work (SOW) templates that include software assurance language; numerous suggestions have already been published for these documents, but final templates need to be published, advertised, distributed, and put into mandatory use
- Give preference to suppliers with a track record of quickly fixing reported flaws
- Implement a scalable supplier assurance process to ensure that critical suppliers are trustworthy and define an evaluation regime that is capable of reviewing vendors‘ actual development processes and rendering a judgment about their ability to produce assured software
- Scan all software that touch the public Internet for vulnerabilities using code analysis tools.

"The Software Supply Chain Integrity Framework: Defining Risks and Responsibilities for Securing Software in the Global Supply Chain," July 21, 2009

SAFECode
Software Assurance Forum for Excellence in Code
Driving Security and Integrity

The Software Supply Chain Integrity Framework

Defining Risks and Responsibilities for Securing Software in the Global Supply Chain

July 21, 2009

EDITOR
Stacy Simpson, SAFECode

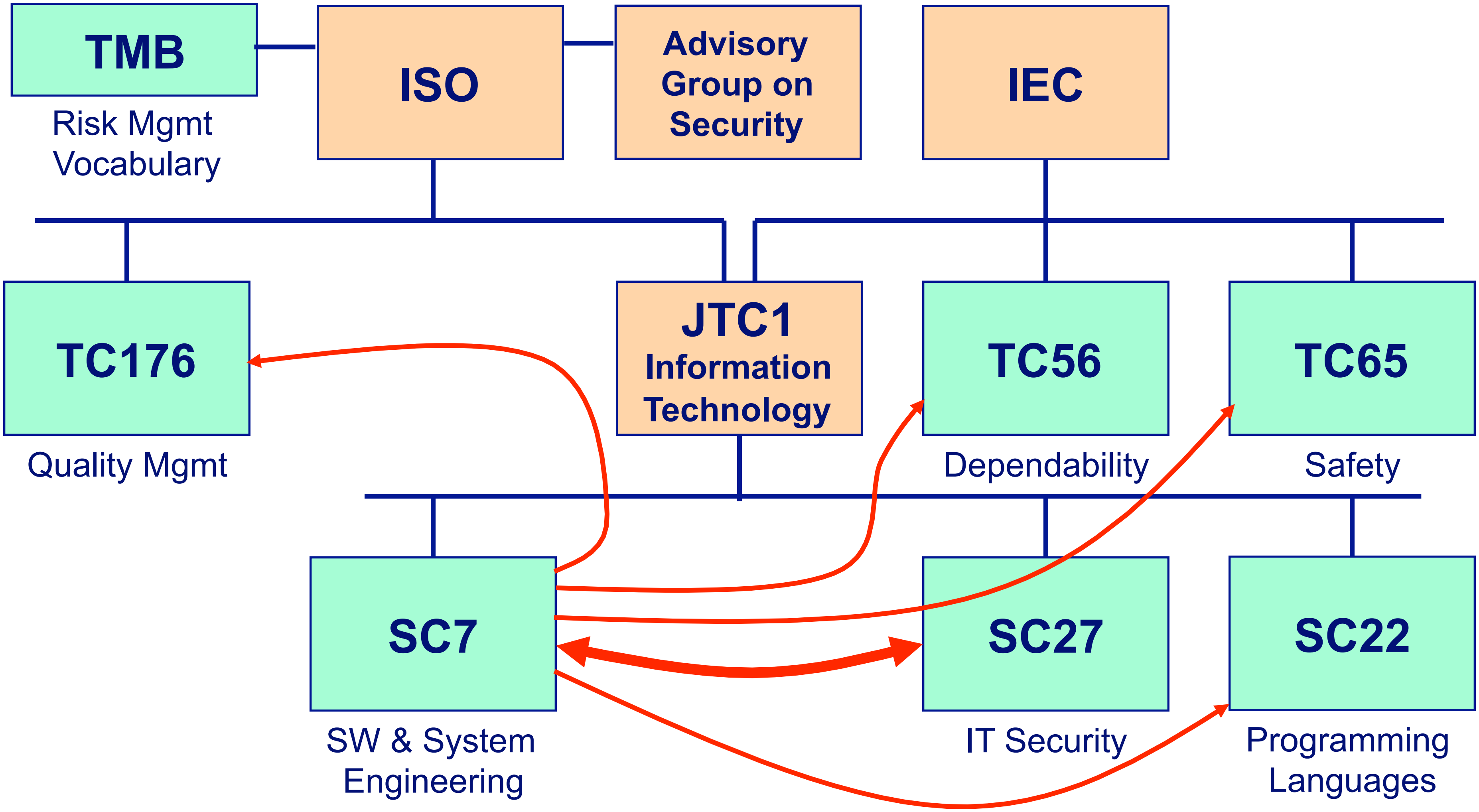
CONTRIBUTORS
Dan Reddy, EMC
Brad Minnis, Juniper Networks
Chris Fagan, Microsoft Corp.
Cheri McGuire, Microsoft Corp.
Paul Nicholas, Microsoft Corp.
Diego Baldini, Nokia
Gunter Bitz, SAP
Yuecel Karabulut, SAP
Gary Phillips, Symantec

As the software industry has become increasingly globalized, a concern has risen over the possibility that an IT solution could be compromised by the intentional insertion of malicious code into the solution's software during its development or maintenance, which is often referred to as a supply chain attack.

Vendors are taking action to mitigate supply chain risk by applying software integrity practices - the collection of processes and controls that enable a vendor to deliver customers a product that is uncompromised, thereby containing only what the vendor intends.

- This 11-page paper outlines an industry-driven framework for analyzing and describing the efforts of software suppliers to mitigate the potential that software could be intentionally compromised during its sourcing, development or distribution.
  - This is released by The Software Assurance Forum for Excellence in Code (SAFECode), a non-profit organization dedicated to increasing trust in information and communications technology products and services through the advancement of effective software assurance methods.
  - It was jointly developed by SAFECode's members, which include EMC Corporation, Juniper Networks, Inc., Microsoft Corp., Nokia, SAP AG and Symantec Corp.
  - Industry members have come together to establish a common framework for ensuring the integrity of software through the global supply chain. This framework will serve the foundation for subsequent work aimed at identifying and analyzing software integrity best practices and represents a critical step forward in the industry's efforts to advance software assurance.
- A full copy of "The Software Supply Chain Integrity Framework: Defining Risks and Responsibilities for Securing Software in the Global Supply Chain" is available for free download at http://www.safecode.org/publications/SAFECode_Supply_Chain0709.pdf

SwA Concerns of Int’l Standards Organizations



TMB

Risk Mgmt Vocabulary

ISO

Advisory Group on Security

IEC

TC176

Quality Mgmt

JTC1 Information Technology

TC56

Dependability

TC65

Safety

SC7

SW & System Engineering

SC27

IT Security

SC22

Programming Languages

Homeland Security

* DHS NCSD has membership on SC7, SC27 & IEEE S2ESC leveraging Liaisons in place or requested with other committees

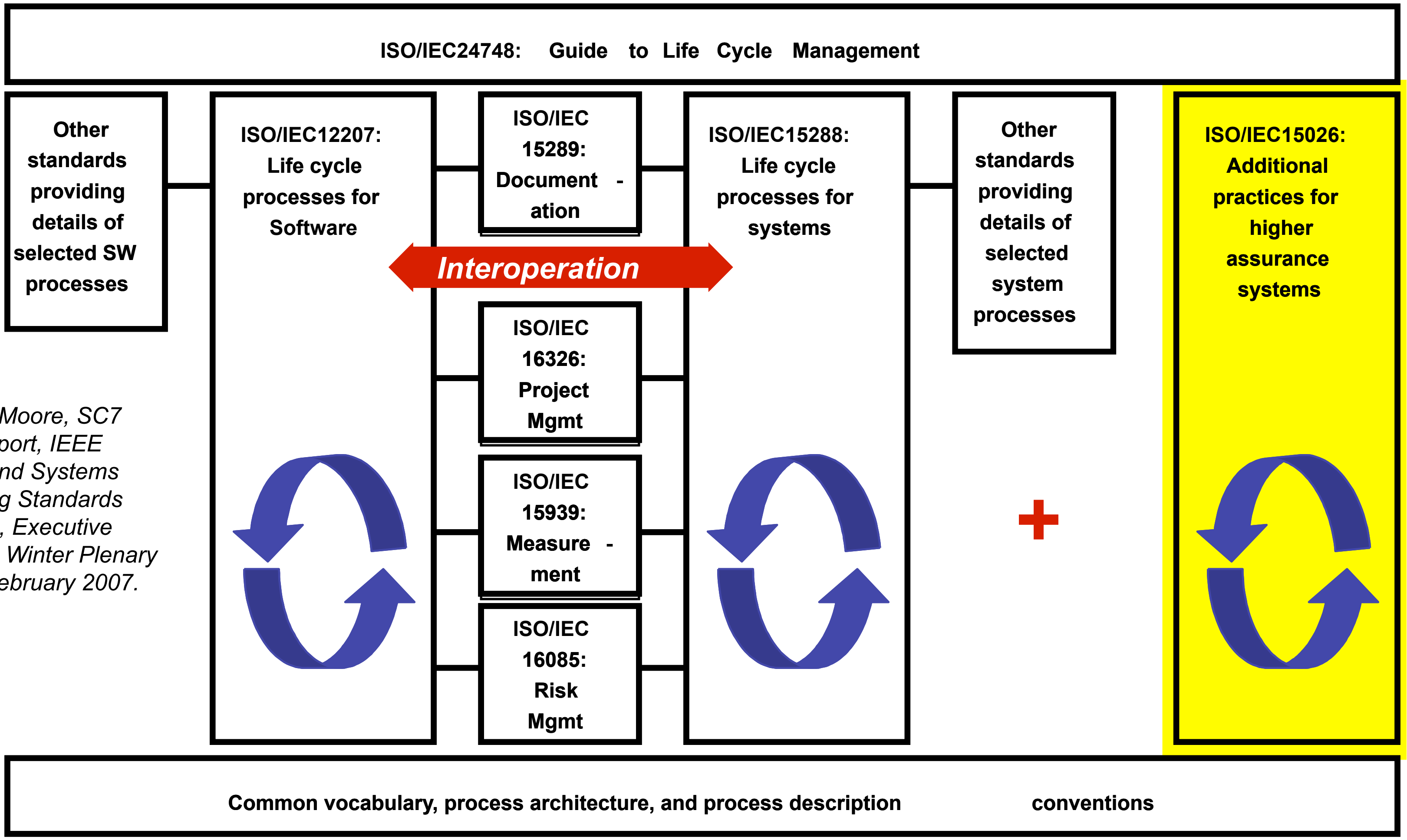
Scope of ISO/IEC JTC1 SC7 Software and Systems Engineering: ISO/IEC 15026 “Systems and Software Assurance”

“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 cycles.”

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

US federal government & suppliers working to ensure consistency with related, evolving Systems and Software Assurance guidelines

ISO/IEC/IEEE 15026, System and Software Assurance



ISO/IEC24748: Guide to Life Cycle Management

Other standards providing details of selected SW processes

ISO/IEC12207: Life cycle processes for Software

ISO/IEC 15289: Document - ation

Interoperation

ISO/IEC 16326: Project Mgmt

ISO/IEC 15939: Measure - ment

ISO/IEC 16085: Risk Mgmt

ISO/IEC15288: Life cycle processes for systems

Other standards providing details of selected system processes

+

ISO/IEC15026: Additional practices for higher assurance systems

Common vocabulary, process architecture, and process description conventions

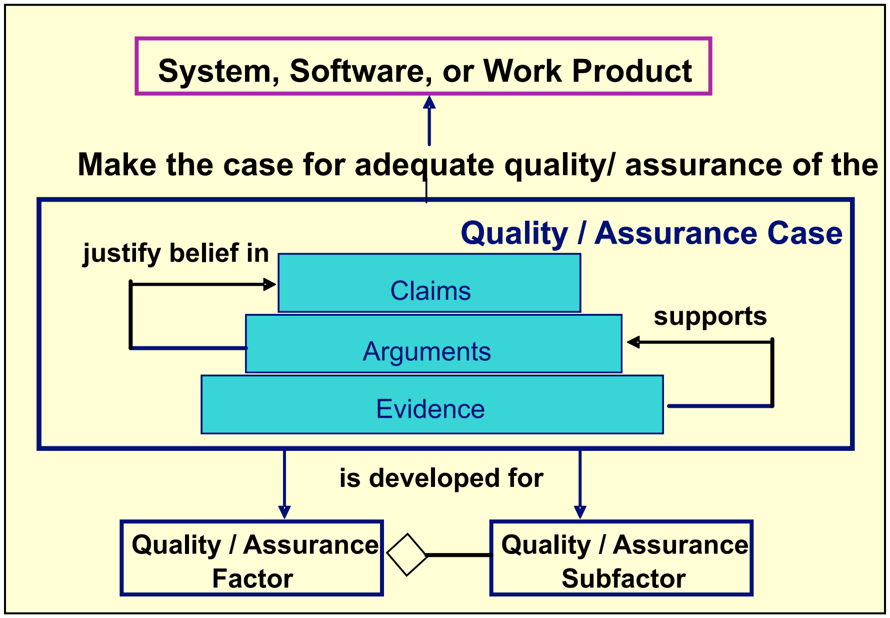
Source: J. Moore, SC7 Liaison Report, IEEE Software and Systems Engineering Standards Committee, Executive Committee Winter Plenary Meeting, February 2007.

"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



System, Software, or Work Product
Make the case for adequate quality/ assurance of the
Quality / Assurance Case
justify belief in
Claims
supports
Arguments
Evidence
is developed for
Quality / Assurance Factor
Quality / Assurance Subfactor

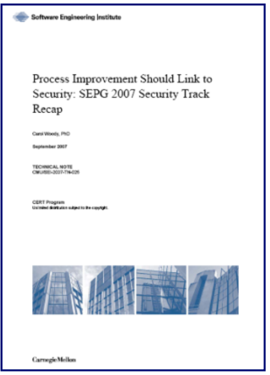
- **Sub-parts**
  - A high level summary
  - Justification that product or service is acceptably safe, secure, or dependable
  - Rationale for claiming a specified level of safety and security
  - Conformance with relevant standards & regulatory requirements
  - The configuration baseline
  - Identified 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

Process Improvement Should Link to Security: SEPG 2007 Security Track Recap

http://www.sei.cmu.edu/publications/documents/07.reports/07tn025.html



Software Engineering Institute

Process Improvement Should Link to Security: SEPG 2007 Security Track Recap

Carol Woody, PhD

September 2007

TECHNICAL NOTE
CMU/SEI-2007-TN-025

CERT Program
Unlimited distribution subject to the copyright.

Carnegie Mellon

Table of Contents

Enhance "Assurance" Considerations:
Leveraging CMM-based Process Improvement

Determine how "assurance" has been factored into suppliers' process capabilities

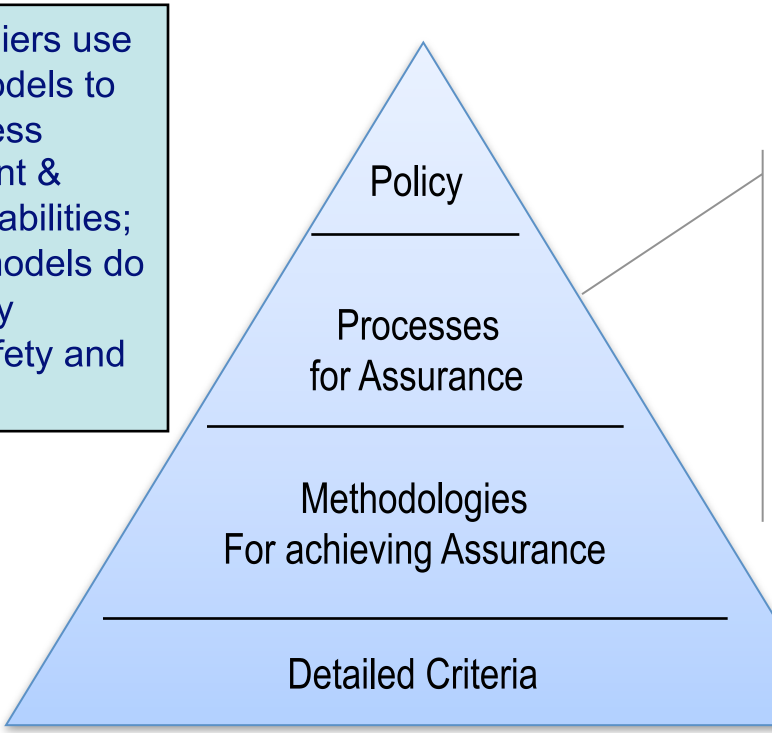
- **An infrastructure for safety & security is established and maintained.**
  1. Ensures Safety and Security Competency within the Workforce;
  2. Establishes a Qualified Work Environment (including the use of qualified tools);
  3. Ensures Integrity of Safety and Security Information;
  4. Monitors Operations and Report Incidents (relative to the deployed environment);
  5. Ensures Business Continuity.
- **Safety & security risks are identified and managed.**
  6. Identifies Safety and Security Risks;
  7. Analyzes and Prioritizes Risks relative to Safety and Security;
  8. Determines, Implements, and Monitors the associated Risk Mitigation Plan.
- **Safety & security requirements are satisfied.**
  9. Determines Regulatory Requirements, Laws, and Standards;
  10. Develops and Deploys Safe and Secure Products and Services;
  11. Objectively Evaluates Products (using safety and security criteria);
  12. Establish Safety and Security Assurance Arguments (with supporting evidence).
- **Activities/products are managed to achieve safety & security requirements/objectives.**
  13. Establishes Independent Safety and Security Reporting;
  14. Establishes a Safety and Security Plan;
  15. Selects and Manages Suppliers, Products, and Services using safety and security criteria;
  16. Monitors and Controls Activities and Products relative to safety and security requirements.

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

Source for "Assurance" enhanced processes: *U.S. DoD and FAA joint project on Safety and Security Extensions for Integrated Capability Maturity Models, September 2004,* at http://www.faa.gov/about/office_org/headquarters_offices/aio/documents/media/SafetyandSecurityExt-FINAL-web.pdf

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

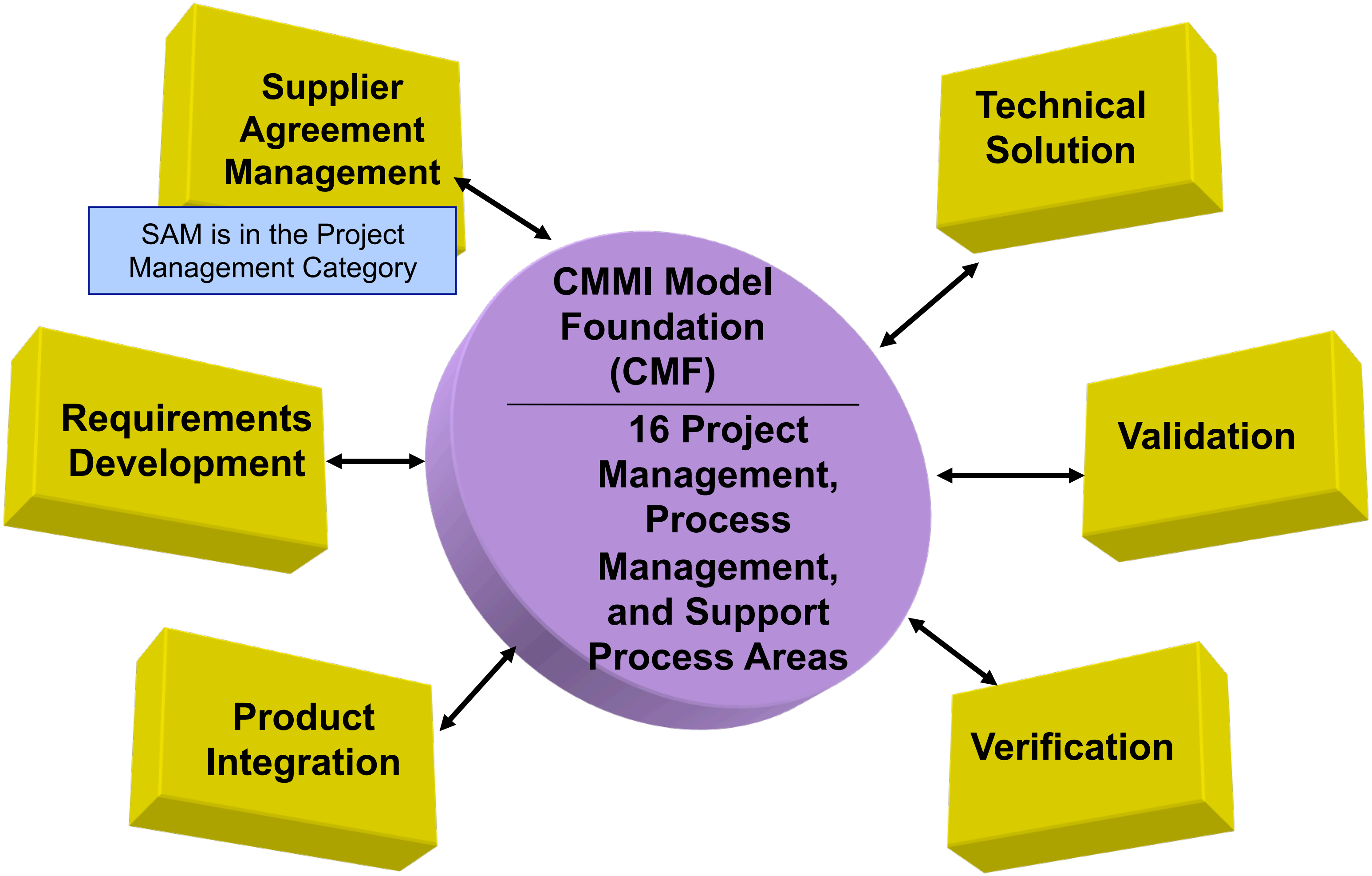
Experience gained for “Assurance” enhanced processes in *U.S. DoD and FAA joint project on Safety and Security Extensions for Integrated Capability Maturity Models, September 2004,* at SwA Community Resources and Information Clearinghouse - see https://buildsecurityin.us-cert.gov/swa/downloads/SafetyandSecurityExt-Sep2004.pdf

Other Assurance Maturity Models have been released in 2009:

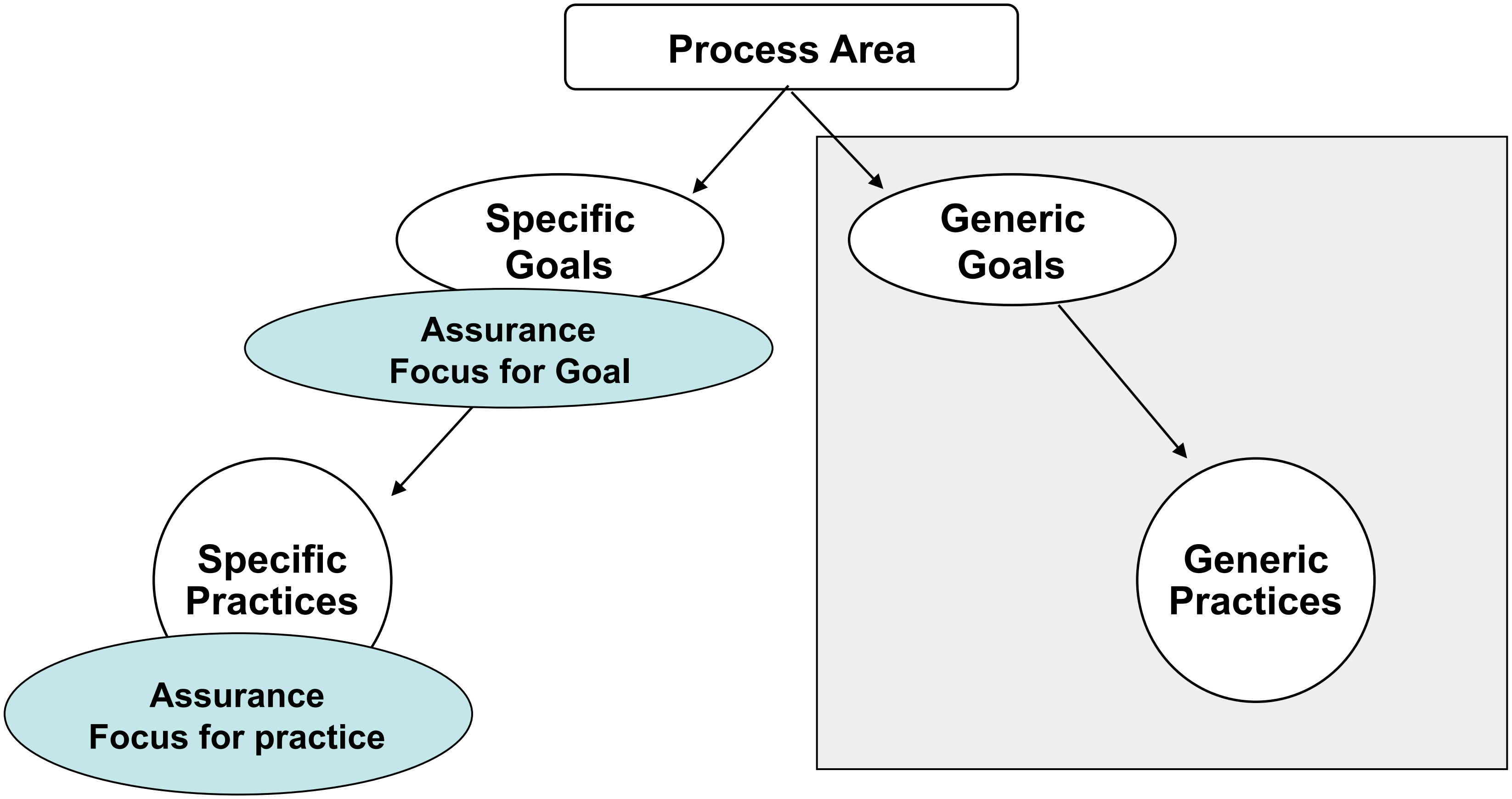
The Building Security In Maturity Model (BSIMM) helps organizations plan software security initiatives http://www.bsi-mm.com/

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/

Assurance for Capability Maturity Model Integration (CMMI)® -- CMMI-DEV v1.2


Supplier Agreement Management
SAM is in the Project Management Category
Technical Solution
Requirements Development
CMMI Model Foundation (CMF)
16 Project Management, Process Management, and Support Process Areas
Validation
Product Integration
Verification

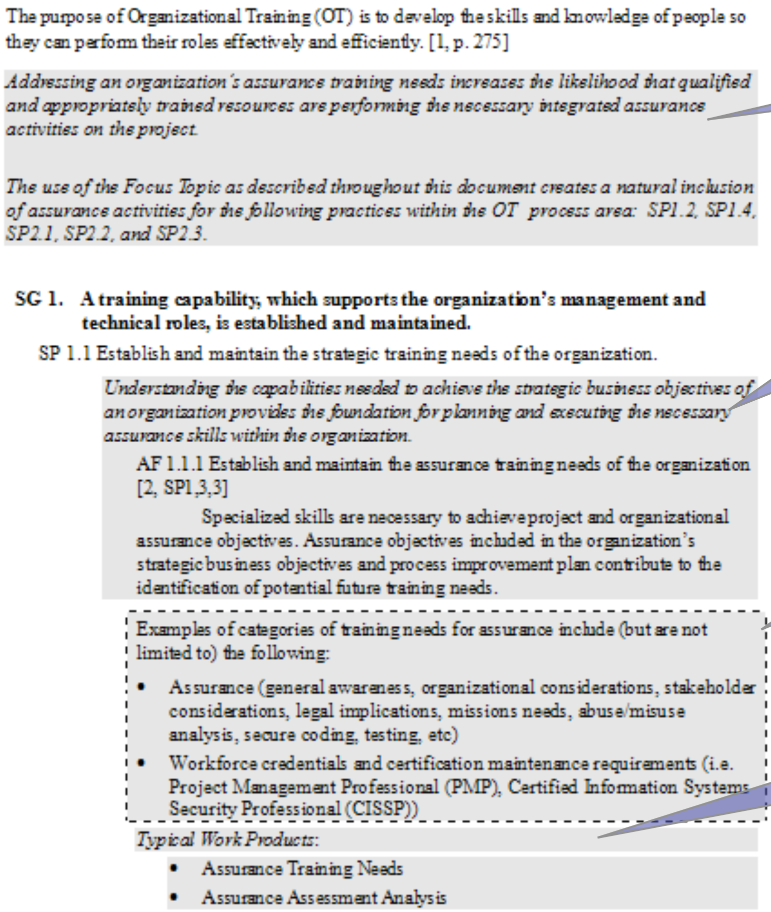
Assurance For CMMI Identifies The Assurance Thread for CMMI-DEV



SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

Assurance Focus For CMMI®



The purpose of Organizational Training (OT) is to develop the skills and knowledge of people so they can perform their roles effectively and efficiently. [1, p. 275]

Addressing an organization's assurance training needs increases the likelihood that qualified and appropriately trained resources are performing the necessary integrated assurance activities on the project.

The use of the Focus Topic as described throughout this document creates a natural inclusion of assurance activities for the following practices within the OT process area: SP1.2, SP1.4, SP2.1, SP2.2, and SP2.3.

SG 1. A training capability, which supports the organization's management and technical roles, is established and maintained.

SP 1.1 Establish and maintain the strategic training needs of the organization.

Understanding the capabilities needed to achieve the strategic business objectives of an organization provides the foundation for planning and executing the necessary assurance skills within the organization.

AF 1.1.1 Establish and maintain the assurance training needs of the organization [2, SP1,3,3]

Specialized skills are necessary to achieve project and organizational assurance objectives. Assurance objectives included in the organization's strategic business objectives and process improvement plan contribute to the identification of potential future training needs.

Examples of categories of training needs for assurance include (but are not limited to) the following:

- Assurance (general awareness, organizational considerations, stakeholder considerations, legal implications, missions needs, abuse/misuse analysis, secure coding, testing, etc)
- Workforce credentials and certification maintenance requirements (i.e. Project Management Professional (PMP), Certified Information Systems Security Professional (CISSP))

Typical Work Products:

- Assurance Training Needs
- Assurance Assessment Analysis

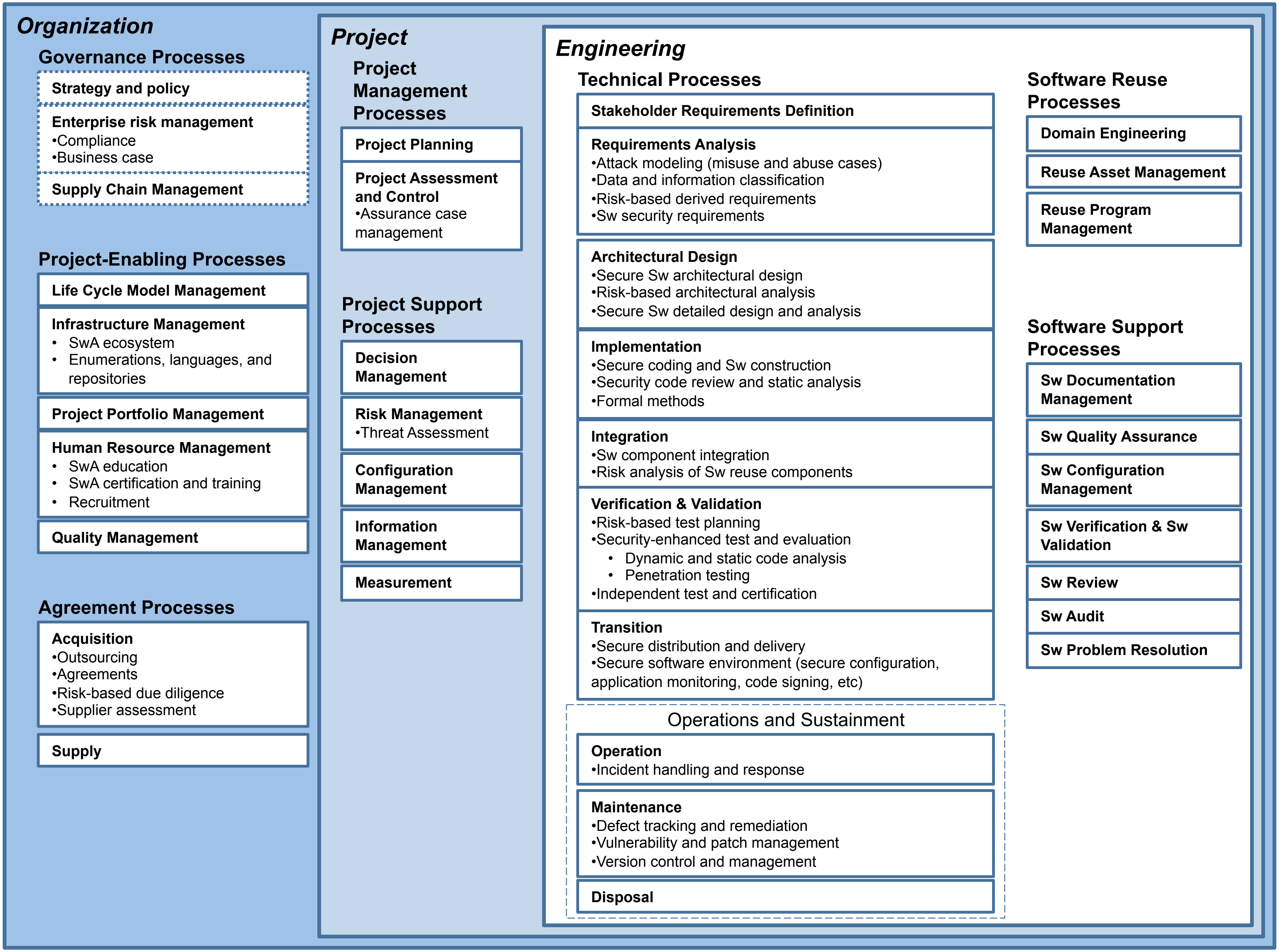
Context of Assurance for the PA

Assurance practice aligned with existing CMMI® specific practice

Supporting examples, sub practices, etc that clarify the Assurance practice

Typical Work Products

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

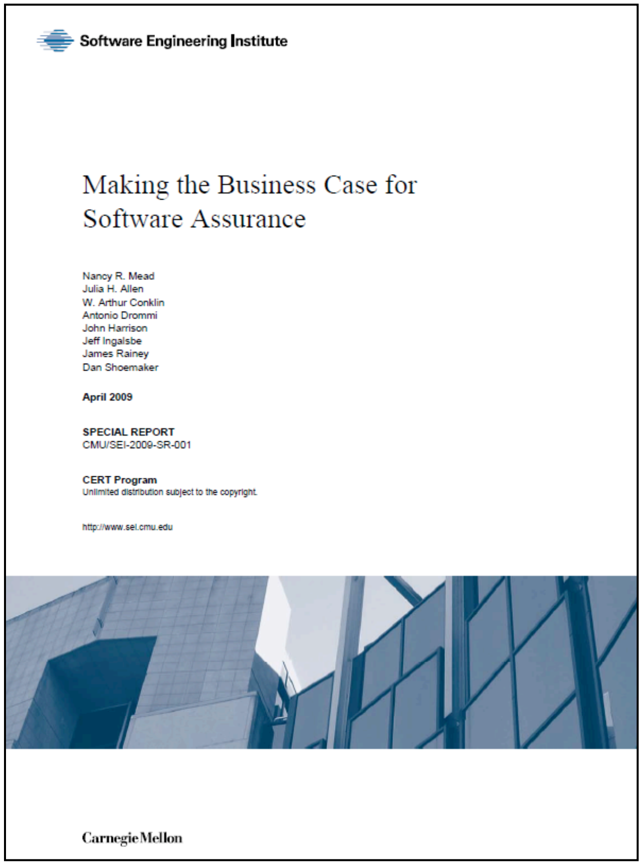
Sw Audit

Sw Problem Resolution

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

CarnegieMellon

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)

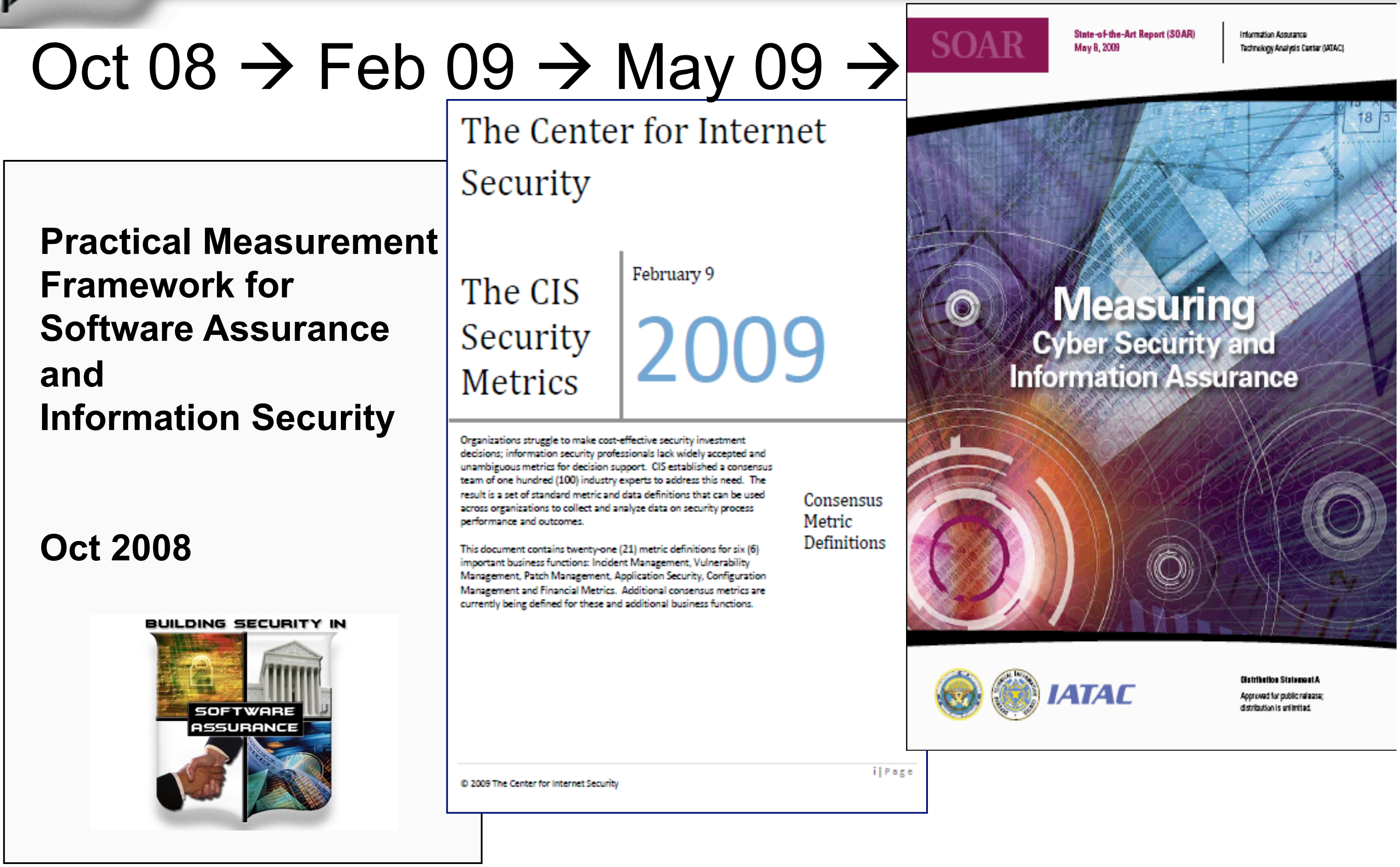
- A listing of resources has been published on the SwA web site targeting primary stakeholder groups: Executive, Developer/Vendor/Supplier, Buyer/Acquirer
  - Sample SwA goals and questions lists to be used to define measures
  - Sources of measurable requirements, such as NIST documents
  - Articles on related subjects, including SwA measurement, security measurement, and software security measurement
  - Useful links
  - Measures library

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

BUILDING SECURITY IN

SOFTWARE ASSURANCE

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

© 2009 The Center for Internet Security

i | Page

SOAR

State-of-the-Art Report (SOAR)
May 8, 2009

Information Assurance
Technology Analysis Center (IATAC)

Measuring Cyber Security and Information Assurance

IATAC

Distribution Statement A

Approved for public release;
distribution is unlimited.

Sponsored by
DHS National Cyber Security Division/US-CERT

NIST
National Institute of
Standards and Technology

National Vulnerability Database

automating vulnerability management, security measurement, and compliance checking

National Vulnerability Database (NVD) Version 2.2 -- http://nvd.nist.gov/

- NVD is the U.S. government repository of standards based vulnerability management data represented using the Security Content Automation Protocol (SCAP).
- This data enables automation of vulnerability management, security measurement, & compliance.
- NVD includes databases of security checklists, security related software flaws, misconfigurations, product names, and impact metrics. NVD supports the Information Security Automation Program.

Federal Desktop Core Configuration settings (FDCC)

- NVD contains content (and pointers to tools) for performing configuration checking of systems implementing the FDCC using the Security Content Automation Protocol (SCAP).
- FDCC Checklists are available to be used with SCAP FDCC Capable Tools -- available via NVD.

NVD Primary Resources

- Vulnerability Search Engine (CVE software flaws and CCE misconfigurations)
- National Checklist Program (automatable security configuration guidance in XCCDF and OVAL)
- SCAP (program and protocol that NVD supports) and SCAP Compatible Tools
- SCAP Data Feeds (CVE, CCE, CPE, CVSS, XCCDF, OVAL)
- Product Dictionary (CPE) and Impact Metrics (CVSS)
- Common Weakness Enumeration (CWE)

Standard Enumerations for Addressing Common Weaknesses and Common Attack Patterns

- Common Weakness Enumeration (CWE) initiative [http://cwe.mitre.org/] and the Common Attack Pattern Enumeration and Classification (CAPEC) [http://capec.mitre.org/] have been sponsored by DHS NCSD:
  - To more effectively understand their risk exposure, consumers need to understand exploitable weaknesses in software before & after put into use.
  - These are standard enumerations and community knowledge resources.
  - These enable consumers to be better informed about the resilience and security of software we acquire and use.
- As a standard enumeration, CWE provides a unified, measurable set of exploitable software weaknesses that now enables more effective discussion, description, selection and use of software security tools and services that can find these weaknesses in source code (with one intent to discover them before the code is put into use).
- CAPEC provides a publicly available catalog of attack patterns along with a comprehensive schema and classification taxonomy; used to better ensure that software functions correctly, even under abnormal and hostile conditions.

Standard Enumerations for Addressing Common Weaknesses and Common Attack Patterns

- CWE is referenced in the National Vulnerability Database http://nvd.nist.gov/nvd.cfm) with the majority linked with CVEs; listed as real-world examples of specific weaknesses,
- CWE provides a foundation for many aspects of software assurance efforts.
  - CWE version 1.0 was publicly available August 2008.
  - CWE Version 1.3 is now available with 762 entreis; more consistent mitigations for 35 entries, especially the Top 25; usage of a more established vocabulary in the names and descriptions of 39 entries; updated relationships for 89 entries, especially the OWASP Top Ten view and the CWE-703 pillar in the Research View; improved labeling of good and bad code blocks in demonstrative examples; and changes to 183 total entries.
  - A detailed report is available that lists specific changes between Version 1.2 and Version 1.3. The CWE Top 25 document has been updated to reflect the changes in the mitigations.
- CWE & CAPEC are important to our community efforts focused on mitigating risks attributable to exploitable weaknesses in software before software is put into use.
- CWE is not currently part of the Security Content Automation Protocol (SCAP). NVD is using CWE as a classification mechanism that differentiates CVEs by the type of vulnerability they represent.
- **NVD (as of 13 May 2009) contains:**
  - 36905 CVE Vulnerabilities **CVE Publication rate:** 16 vulnerabilities / day
  - 142 Checklists
  - 173 US-CERT Alerts
  - 2330 US-CERT Vuln Notes
  - 2517 OVAL Queries

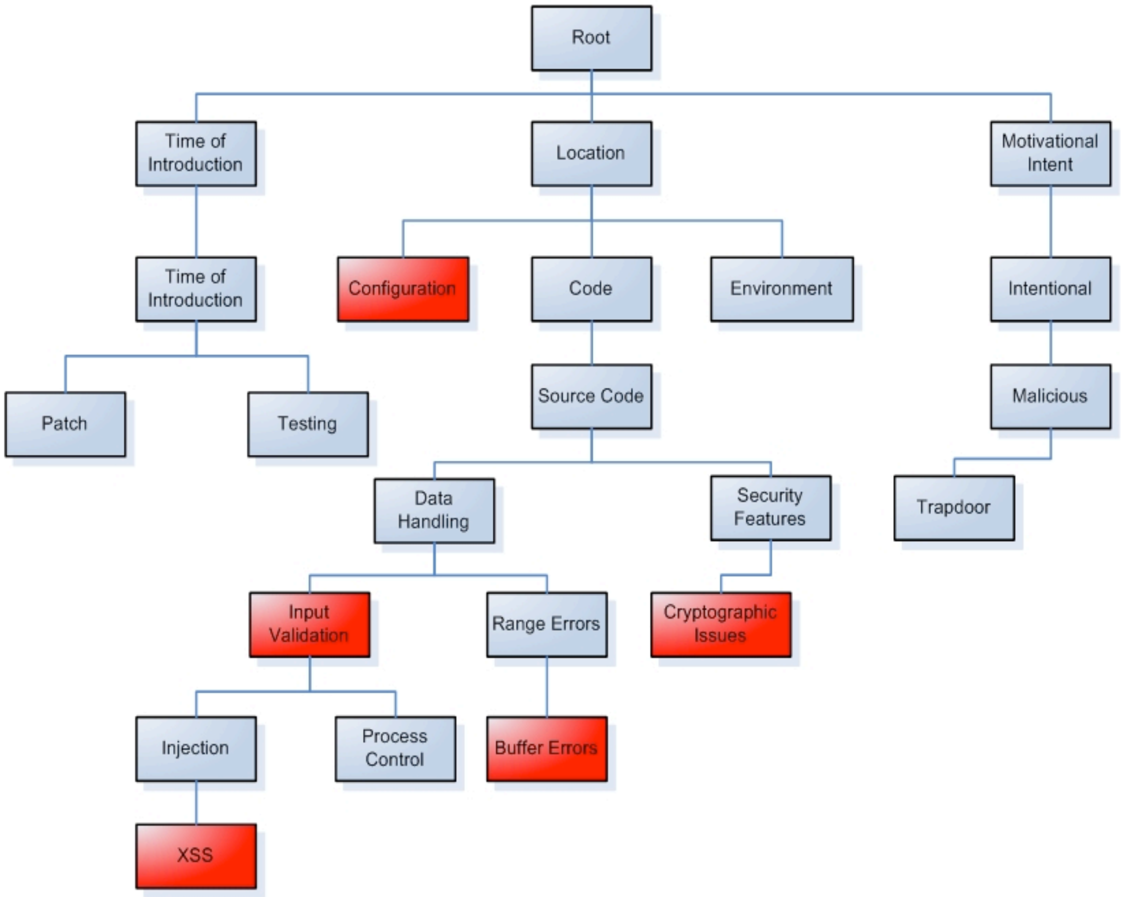
CWE used with CVE scoring

Common Weakness Enumeration (CWE) specification provides a common language of discourse for discussing, finding and dealing with the causes of software security vulnerabilities as they are found in code, design, or system architecture.

Each individual CWE represents a single vulnerability type. CWE is maintained by the MITRE Corporation with support from the National Cyber Security Division (DHS). A detailed CWE list is currently available at the MITRE website; this list provides a detailed definition for each individual CWE.

All individual CWEs are held within a hierarchical structure that allows for multiple levels of abstraction. CWEs located at higher levels of the structure (i.e. Configuration) provide a broad overview of a vulnerability type and can have many children CWEs associated with them. CWEs at deeper levels in the structure (i.e. Cross Site Scripting) provide a finer granularity and usually have fewer or no children CWEs. The image to the right represents a portion of the overall CWE structure, the red boxes represent the CWEs being used by NVD.

Portion of CWE Structure



NVD integrates CWE into the scoring of CVE vulnerabilities by providing a cross section of the overall CWE structure. NVD analysts score CVEs using CWEs from different levels of the hierarchical structure. This cross section of CWEs allows analysts to score CVEs at both a fine and coarse granularity, which is necessary due to the varying levels of specificity possessed by different CVEs.

See http://measurablesecurity.mitre.org/ for a better understanding of how common enumerations link together

Key Practices for Mitigating the Most Egregious Exploitable Software Weaknesses

Software Assurance Pocket Guide Series: Development,
Volume II, Version 1.3, May 24, 2009 (Draft)

- ***Table 1 - Top 25 Common Weakness Enumeration (CWE)***
- ***Table 2 – CWEs and Their Related Attack Patterns and Mission/Business Risks***

- **Tables 3-5 – Prevention and Mitigation Practices listed by lifecycle phases**
  - **Requirements, Architecture , and Design Phases**
  - **Build, Compilation, Implementation, Testing, and Documentation Phases**
  - **Installation, Operation, and System Configuration Phases**

Table 1 – Top 25 Common Weakness Enumeration (CWE)

Insecure Interaction Between Components These weaknesses are related to insecure ways in which data is sent and received between separate components, modules, programs, processes, threads, or systems.

CWE-20:	Improper Input Validation.
CWE-116:	Improper Encoding or Escaping of Output.
CWE-89:	Failure to Preserve SQL Query Structure (aka ‘SQL Injection’).
CWE-79:	Failure to Preserve Web Page Structure (aka ‘Cross-site Scripting’).
CWE-78:	Failure to Preserve OS Command Structure (aka ‘OS Command Injection’).
CWE-319:	Cleartext Transmission of Sensitive Information
CWE-352:	Cross-Site Request Forgery (CSRF).
CWE-362:	Race Condition.
CWE-209:	Error Message Information Leak.

Risky Resource Management These weaknesses are related to ways in which software does not properly manage the creation, usage, transfer, or destruction of important system resources.

CWE-119:	Failure to Constrain Operations within the Bounds of a Memory Buffer.
CWE-642:	External Control of Critical State Data.
CWE-73:	External Control of File Name or Path.
CWE-426:	Untrusted Search Path.
CWE-94:	Failure to Control Generation of Code (aka ‘Code Injection’).
CWE-494:	Download of Code Without Integrity Check.
CWE-404:	Improper Resource Shutdown or Release.
CWE-665:	Improper Initialization.
CWE-682:	Incorrect Calculation.

Porous Defenses These weaknesses are related to defensive techniques that are often misused, abused, or just plain ignored.

CWE-285:	Improper Access Control (Authorization).
CWE-327:	Use of a Broken or Risky Cryptographic Algorithm.
CWE-259:	Hard-Coded Password.
CWE-732:	Insecure Permission Assignment for Critical Resource.
CWE-330:	Use of Insufficiently Random Values.
CWE-250:	Execution with Unnecessary Privileges.
CWE-602:	Client-Side Enforcement of Server- Side Security.

Table 2 – CWEs and Their Related Attack Patterns and Mission/Business Risks

CWE-89: Failure to Preserve SQL Query Structure (aka ‘SQL Injection’)

» Blind SQL Injection (CAPEC ID:7).
» SQL Injection (CAPEC ID:66).

» Allow execution of malicious/arbitrary code.
» Access or modification of sensitive data and/or Leak information.

CWE-79: Failure to Preserve Web Page Structure (aka ‘Cross-site Scripting’)

» Embedding Scripts (various types, CAPEC IDs: 19, 32, 86).
» Client Network Footprinting (using AJAX/XSS, CAPEC ID:85).
» XSS in IMG Tags (CAPEC ID:91).

» Allow execution of malicious/arbitrary code.
» Escalate privileges.
» Leak information.

CWE-78: Failure to Preserve OS Command Structure (aka ‘OS Command Injection’)

» Argument Injection (CAPEC ID:6).
» Command Delimiters (CAPEC ID:15).
» Exploiting Multiple Input Interpretation Layers (CAPEC ID:43).
» Command Injection (CAPEC ID:88).

» Allow execution of malicious/arbitrary code.
» Modify data and/or Leak information.
» Escalate privileges.

Table 2 – CWEs and Their Related Attack Patterns and Mission/Business Risks

CWE-319: Cleartext Transmission of Sensitive Information

» Passively Sniff/Capture Application Code Bound for Authorized Client (CAPEC ID:65).

» Leak information or Escalate privileges.

CWE-352: Cross-Site Request Forgery (CSRF)

» Cross Site Request Forgery (aka Session Riding , CAPEC ID:62).

» Leak information and/or Modify data or Escalate privileges.

CWE-362: Race Condition

» Leveraging Race Conditions (CAPEC ID:26).
» Leveraging Time-of-Check & Time-of-Use Race Conditions (CAPEC ID:29).

» Escalate privileges.
» Leak information and/or Modify data.
» Allow execution of malicious/arbitrary code.
» Render system unusable (AKA denial of service).

CWE-209: Error Message Information Leak

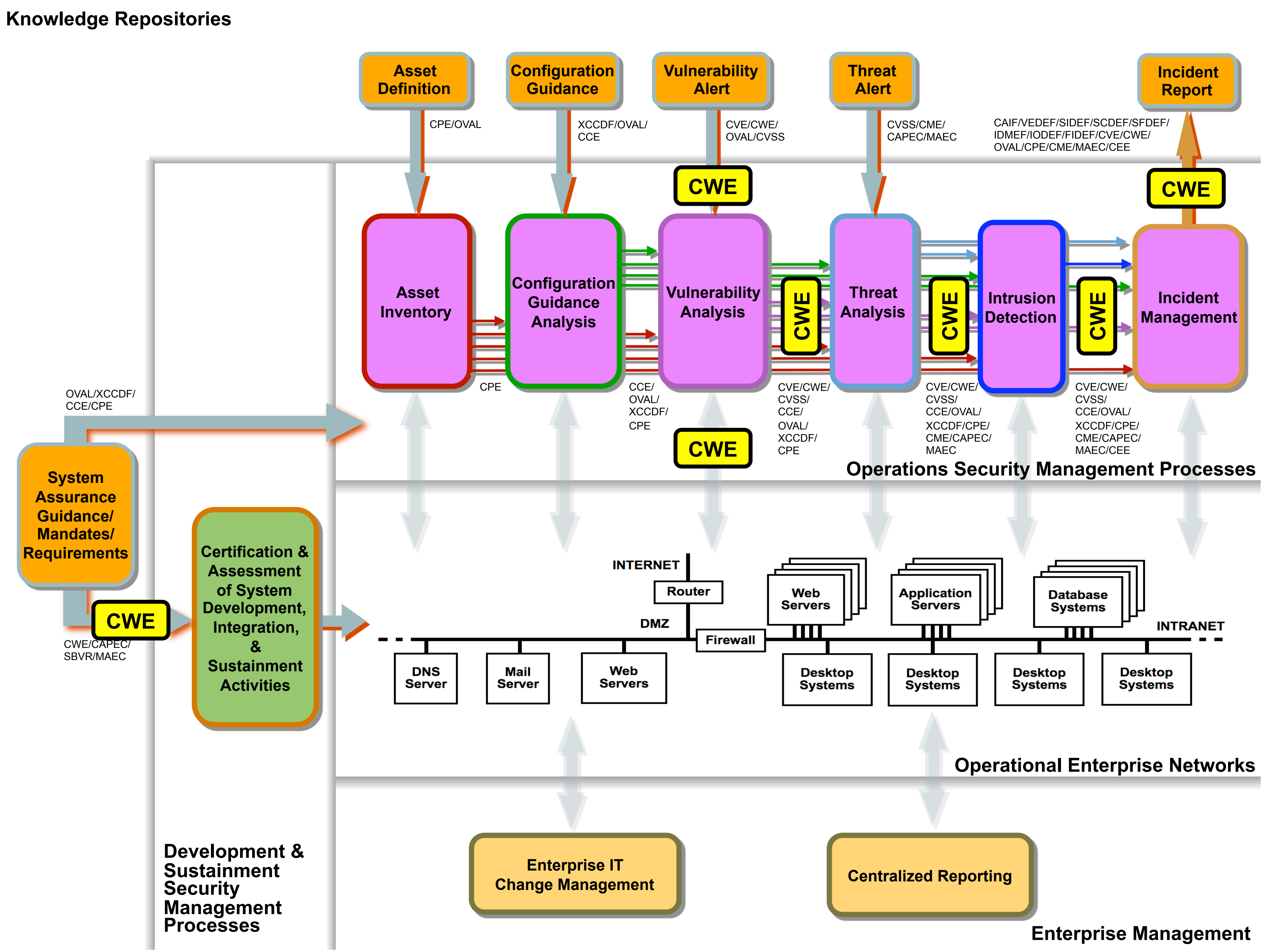
» Blind SQL Injection (CAPEC ID:7).
» Probing an Application Through Targeting its Error Reporting (CAPEC ID:54).

» Leak information and/or Modify data or » Allow execution of malicious/arbitrary code.

CWE-119: Failure to Constrain Operations within the Bounds of a Memory Buffer

» Overflow (various types, CAPEC IDs: 8, 9, 14, 24, 44, 45, 46, 47,100).

» Gain control of the system or Crash the system (denial of service).


Knowledge Repositories
Asset Definition
Configuration Guidance
Vulnerability Alert
Threat Alert
Incident Report
CPE/OVAL
XCCDF/OVAL/ CCE
CVE/CWE/ OVAL/CVSS
CVSS/CME/ CAPEC/MAEC
CAIF/VEDEF/SIDEF/SCDEF/SFDEF/ IDMEF/IODEF/FIDEF/CVE/CWE/ OVAL/CPE/CME/MAEC/CEE
CWE
CWE
Asset Inventory
Configuration Guidance Analysis
Vulnerability Analysis
CWE
Threat Analysis
CWE
Intrusion Detection
CWE
Incident Management
CPE
CCE/ OVAL/ XCCDF/ CPE
CVE/CWE/ CVSS/ CCE/ OVAL/ XCCDF/ CPE
CVE/CWE/ CVSS/ CCE/OVAL/ XCCDF/CPE/ CME/CAPEC/ MAEC
CVE/CWE/ CVSS/ CCE/OVAL/ XCCDF/CPE/ CME/CAPEC/ MAEC/CEE
CWE
Operations Security Management Processes
OVAL/XCCDF/ CCE/CPE
System Assurance Guidance/ Mandates/ Requirements
CWE
CWE/CAPEC/ SBVR/MAEC
Certification & Assessment of System Development, Integration, & Sustainment Activities
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
Development & Sustainment Security Management Processes
Enterprise IT Change Management
Centralized Reporting
Enterprise Management

Software Assurance:
Delivering System Predictability and Reducing Uncertainty

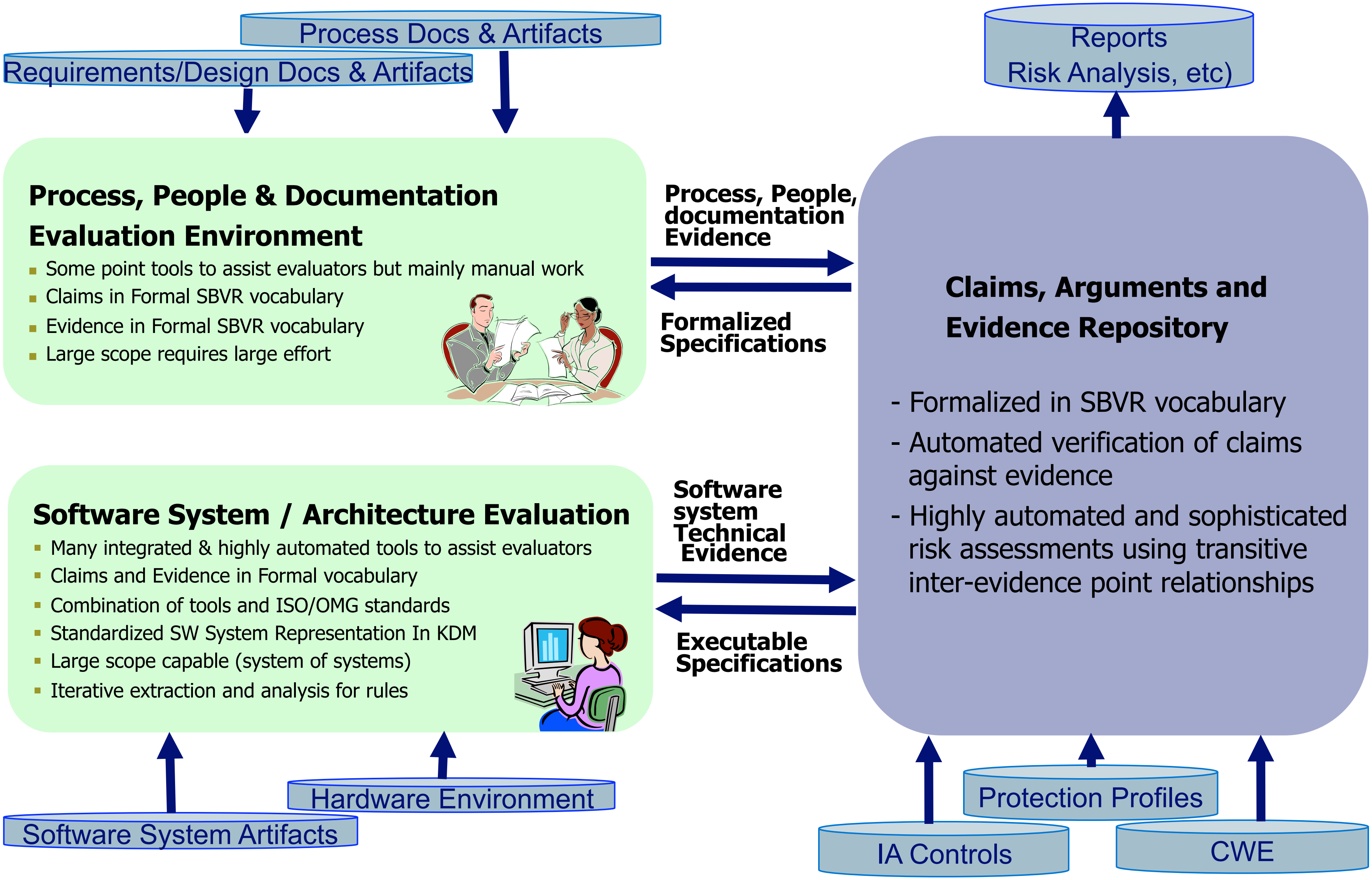
- Software Assurance (SwA) includes processes & practices that:

1. **Specify Assurance Case**
   - Enable supplier to make assurance claims about safety, security and/or dependability of systems, product or services
2. **Obtain Evidence for Assurance Case**
   - Perform assurance assessments to justify claims of meeting a set of requirements through a structure of claims, arguments, and supporting evidence
   - Collect evidence and verifying claims' compliance is complex and costly process
3. **Use Assurance Case to calculate and mitigate risk**
   - Exam non-conformant claims and their evidence to calculate risk and identify course of actions to mitigate it
   - Each stakeholder will have own risk assessment – e.g. security, liability, performance, compliance

SwA processes & practices are moving toward more disciplined, less subjective with more automated, comprehensive tooling and formalized specifications

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

Hardware Environment

Software System Artifacts

Protection Profiles

IA Controls

CWE

Software Assurance Ecosystem:
Turning Challenges into Solutions

- SwA Ecosystem is a formal framework for analysis and exchange of information related to software security and trustworthiness
- Provides a technical environment where formalized claims, arguments and evidence can be brought together with formalized and abstracted software system representations to support high automation and high fidelity analysis.
- Based entirely on international (ISO/IEC/OMG) Open Standards
  - Semantics of Business Vocabulary and Rules (SBVR)
  - Knowledge Discovery Meta-model (KDM)
  - Software Assurance Meta-model (SAM) – work in progress for Assurance Case
    - Software Assurance Evidence Metamodel
    - Software Assurance Claims & Arguments Metamodel
- Architected with a focus on providing fundamental improvements in analysis

Leveraging what we already have through SwA Ecosystem

- Software Assurance Ecosystem enables industry and government to **leverage** and **connect** existing standards, policies, practices, processes and tools, in an affordable and efficient manner
- The key enabler is the Software Assurance (SwA) Ecosystem Infrastructure
  - an open standard-based integrated tooling environment that dramatically reduces the cost of software assurance activities
    - Integrates different communities for a SwA solution:
      - Formal Methods,
      - Reverse Engineering,
      - Static Analysis, and
      - Dynamic Analysis
    - Enables different tool types to interoperate
    - Introduces many new vendors to ecosystem because they each leverage parts of the method/tool chain

BUILDING SECURITY IN

SOFTWARE ASSURANCE

IT/Software Supply Chain Management is a National Security 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
- 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 9-12 March 2010 at MITRE, McLean Virginia
Next SwA Working Group Session 15-17 Dec 2009 at MITRE, McLean VA

BUILDING SECURITY IN
SOFTWARE ASSURANCE

SOFTWARE ASSURANCE FORUM

“Building Security In”

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

Homeland Security

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

Working for Homeland Security

The DHS Office of Cybersecurity and Communications (CS&C) serves as the national focal point for securing cyber space and the nation's cyber assets.

CS&C is actively seeking top notch talent in several areas including:

- Software assurance
- Information technology
- Telecommunications
- Program management
- Public affairs

To learn more about CS&C and potential career opportunities, please visit USAJOBS at www.usajobs.gov .

SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

Homeland Security

Department of Commerce

National Defense

Next SwA Forum 9-12 March 2010 at MITRE, McLean VA

Next SwA Working Group Session 15-17 Dec 2009 at MITRE, McLean VA