

# SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN



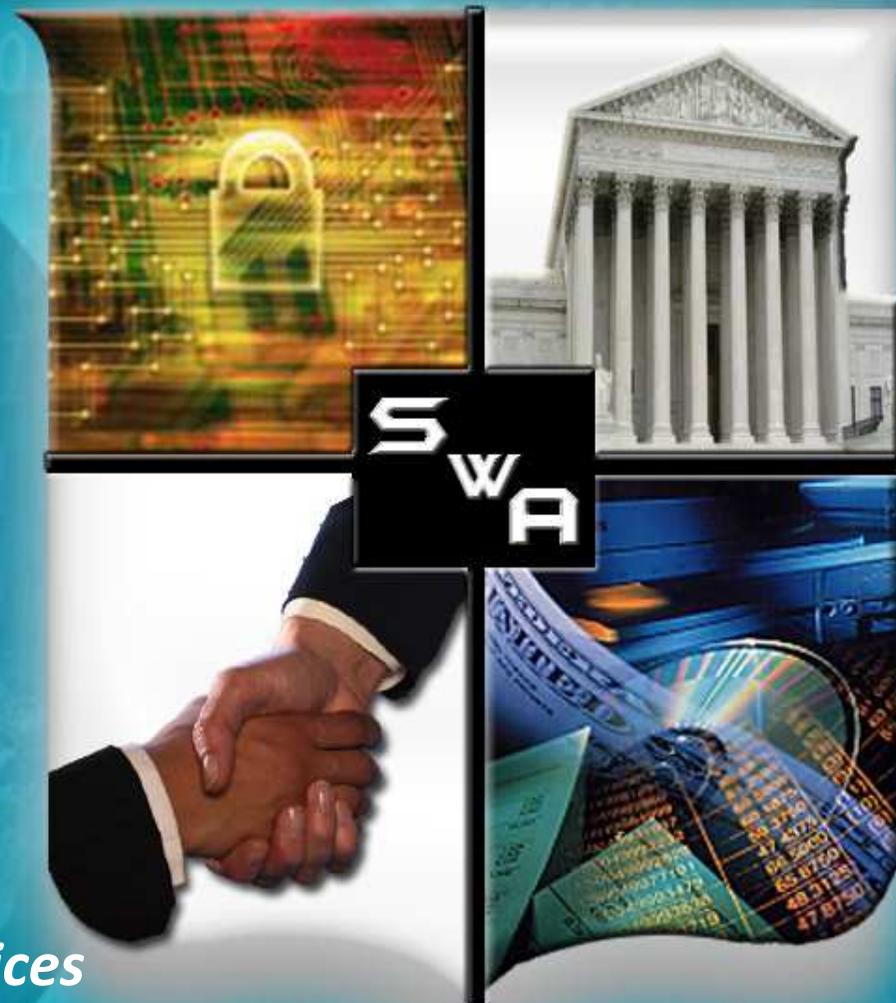
Homeland  
Security



Commerce

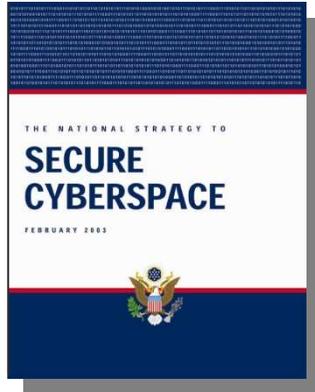


National  
Defense

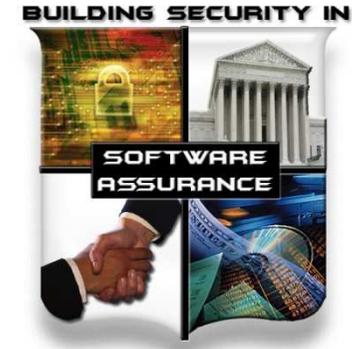


*Game-Changing Tools and Practices*

Next SwA Forum 14-16 Dec 2010 at MITRE in McLean, VA



# Software Assurance



## Game-Changing Tools and Practices

Sept 28, 2010



Homeland  
Security

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

# DHS NCSD Software Assurance (SwA) Program

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

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



# SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN



Homeland  
Security



Commerce



National  
Defense



*Game-Changing Tools and Practices*

Next SwA Forum 14-16 Dec 2010 at MITRE in McLean, VA

# Software Assurance (SwA) – Security Automation

---

10:45am - SwA Panel: Use Cases, Standards and  
Roadmap for Enterprise Security Automation

11:45am - Knowing Your Weaknesses (CWE)

1:30pm - Ranking Your Weaknesses (CWSS)

2:30pm - Understanding How They Attack Your  
Weaknesses (CAPEC)

3:45pm - Sharing Understanding of Malware (MAEC)

4:45pm - Panel on SwA Automation Protocol



# Software Assurance (SwA) – Security Automation

---

## SwA Panel: Use Cases, Standards and Roadmap for Enterprise Security Automation

- Panel Facilitator – Joe Jarzombek, DHS NCSD
- Relevant International Standards – Don Davidson, DoD
- Enterprise Security Automation – Bob Martin, MITRE
- Incident Tracking, Event Management and Threat Analysis: Operational Applications for Automation Protocols – Tom Millar, US-CERT
- Use Cases for Security Automation – Dan Schmidt, NSA and Tim Grance, NIST



# Software Assurance (SwA) – Security Automation

---

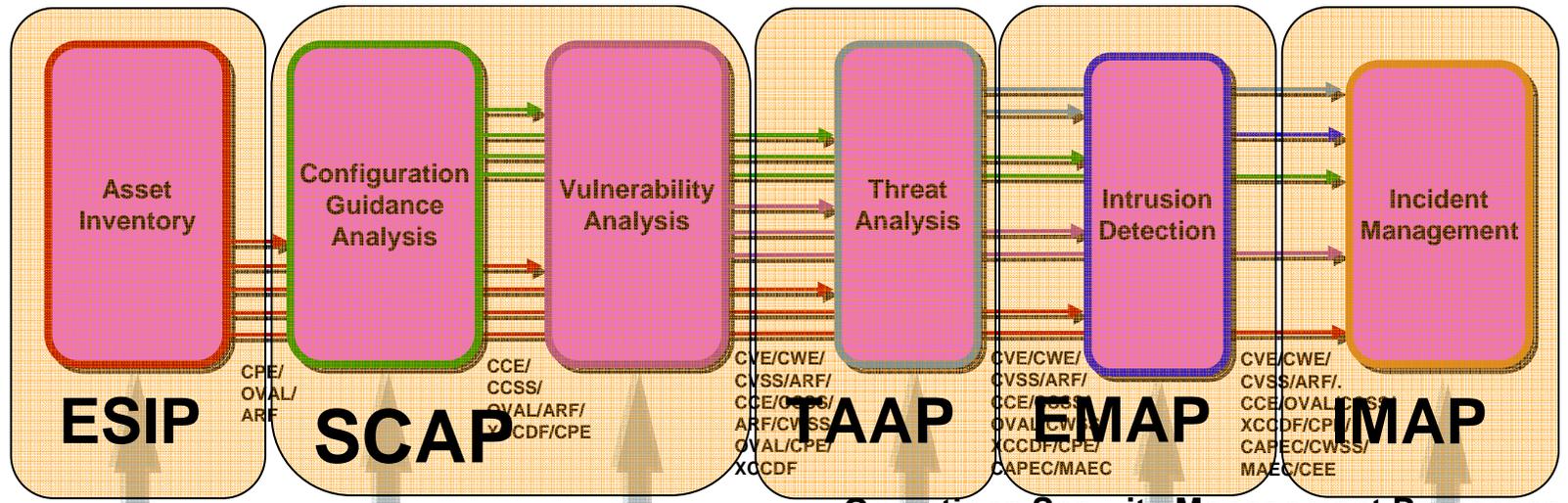
- **Security Content Automation Protocol (SCAP)**
- **Software Assurance Automation Protocol (SwAAP)**
- **Enterprise System Information Protocol (ESIP)**
- **Enterprise Remediation Automation Protocol (ERAP)**
- **Enterprise Compliance Automation Protocol (ECAP)**
- **Event Management Automation Protocol (EMAP)**
- **Incident Tracking and Assessment Protocol (ITAP)**
- **Threat Analysis Automation Protocol (TAAP)**

## Use Cases for Enterprise IT Security



|       | SCAP | SwAAP | ESIP | ERAP | ECAP | EMAP | ITAP | TAAP |
|-------|------|-------|------|------|------|------|------|------|
| CVE   | X    |       |      |      |      |      | X    | X    |
| OVAL  | X    |       |      |      |      |      | X    | X    |
| XCCDF | X    |       |      |      |      |      |      |      |
| CVRF  | X    |       |      |      |      |      |      |      |
| OCIL  | X    |       |      |      |      |      | X    |      |
| CPE   | X    |       | X    |      |      |      | X    | X    |
| CCE   | X    |       |      |      |      |      |      | X    |
| CWE   |      | X     |      |      |      |      |      | X    |
| CAPEC |      | X     |      |      |      | X    | X    | X    |
| MAEC  |      | X     |      |      |      | X    | X    | X    |

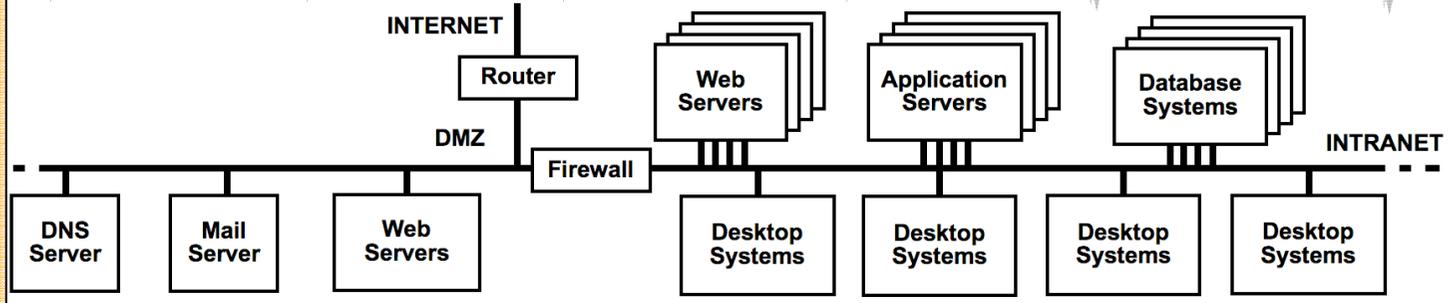




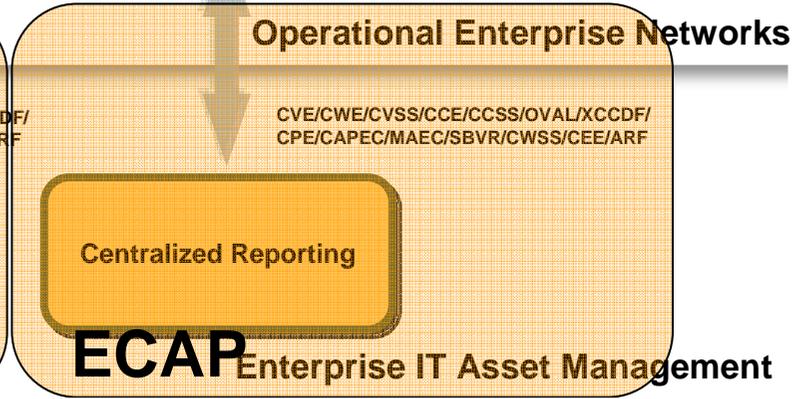
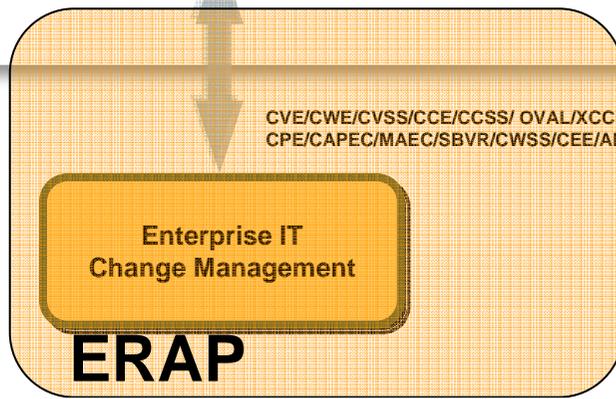
**Operations Security Management Processes**



**Development & Sustainment Security Management Processes**



**Operational Enterprise Networks**



**Enterprise IT Asset Management**

# Software Assurance (SwA) – Security Automation

---

Panel on Software Assurance Automation Protocol

Facilitator: Joe Jarzombek, DHS NCSD

Steve Quinn, NIST

Dan Schmidt, NSA



Homeland  
Security

# Cyberspace

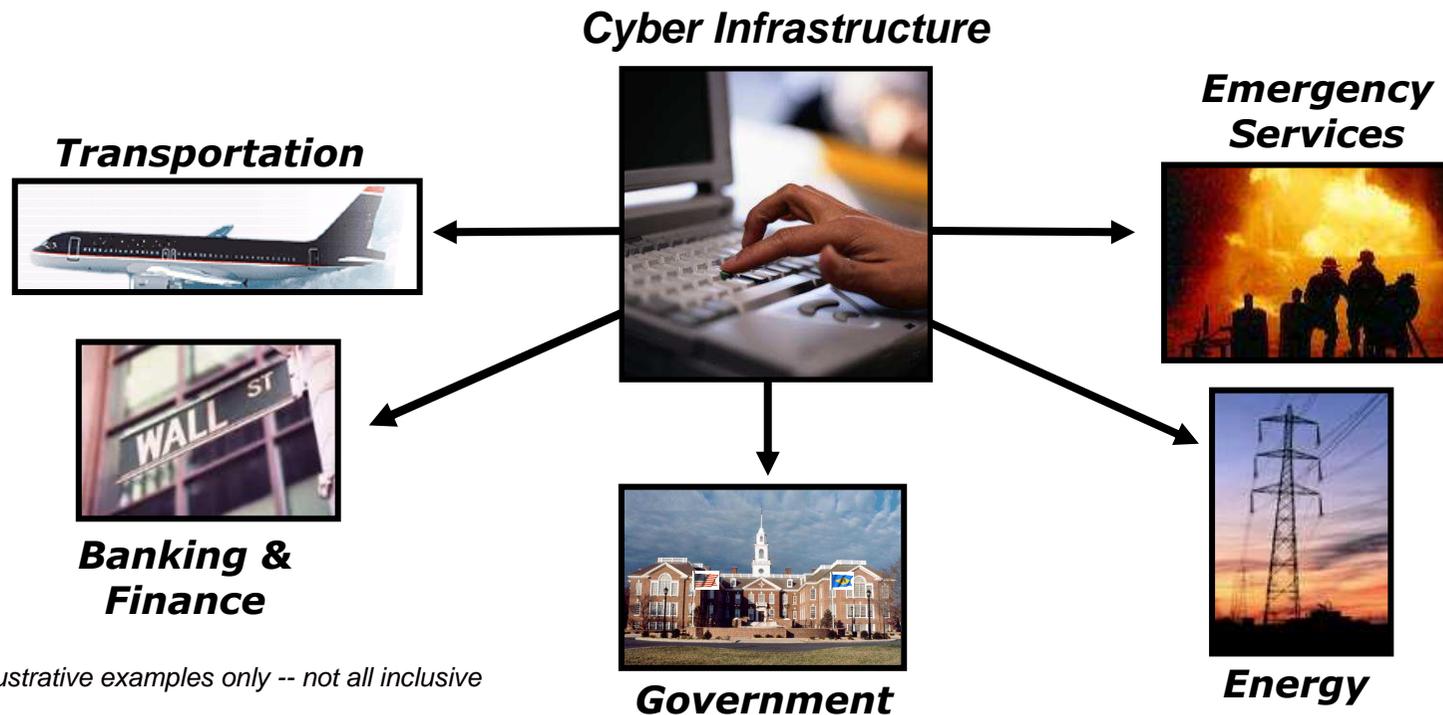
---

- 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

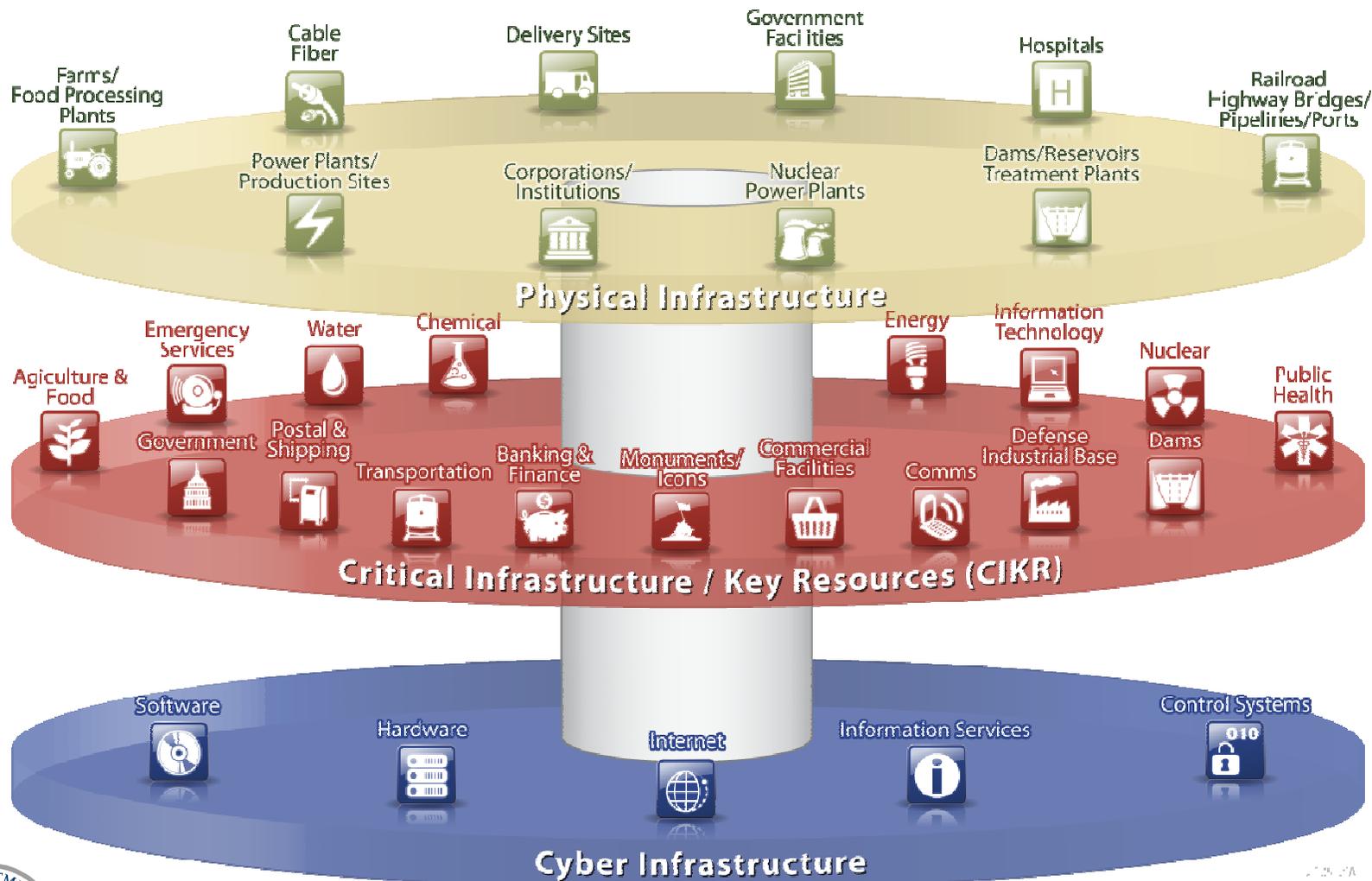


*Illustrative examples only -- not all inclusive*



Homeland  
Security

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



Homeland Security

# Critical Considerations

---



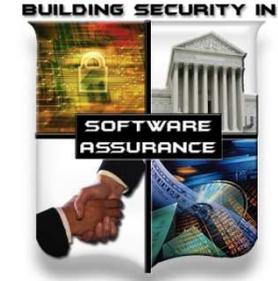
- ▶ Software is the core constituent of modern products and services – it enables functionality and business operations
- ▶ Dramatic increase in mission risk due to increasing:
  - Software dependence and system interdependence (weakest link syndrome)
  - Software Size & Complexity (obscures intent and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



**Homeland  
Security**

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

# 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.
  - Need rating schemes for software products and supplier capabilities



# Security-Enhanced Capabilities: Mitigating Risks to the Enterprise

---



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



**Homeland  
Security**

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

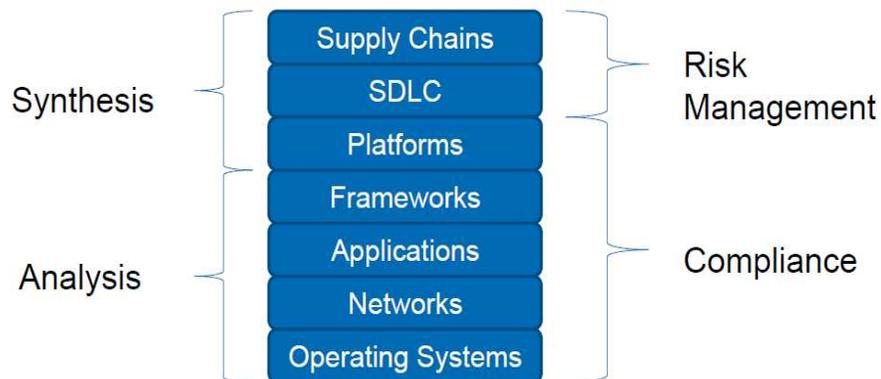
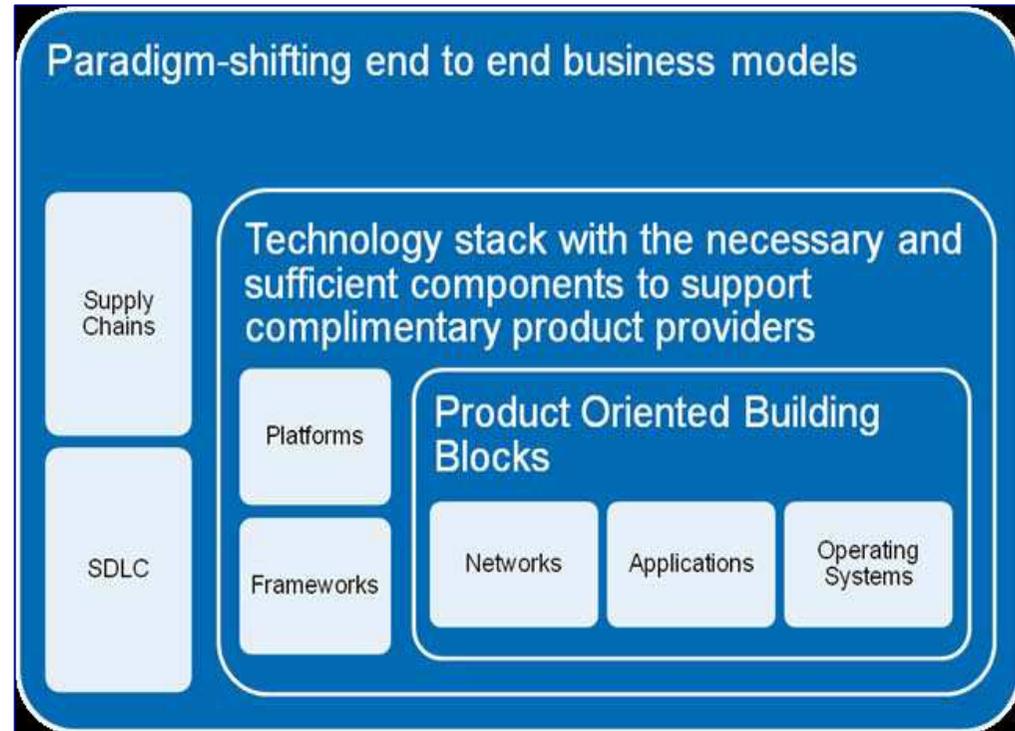
# IT/software security risk landscape is a convergence between “defense in depth” and “defense in breadth”

Enterprise Risk Management and Governance are security motivators

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

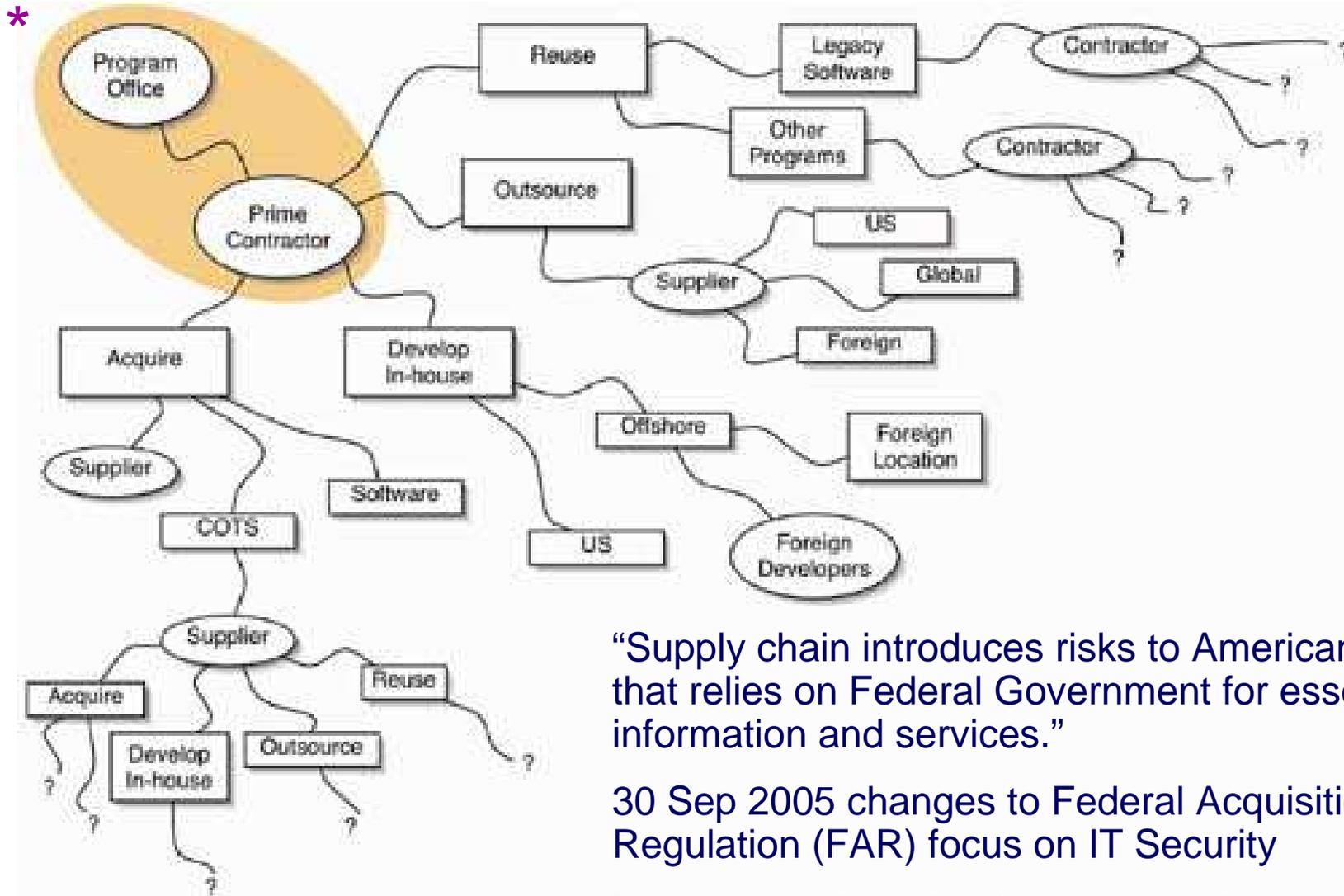
“In the digital age, sovereignty is demarcated not by territorial frontiers but by supply chains.”

– Dan Geer, CISO In-Q-Tel



Software Assurance provides a focus for:

- Secure Software Components,
- Security in the Software Life Cycle and
- Software Supply Chain Risk Management



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

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

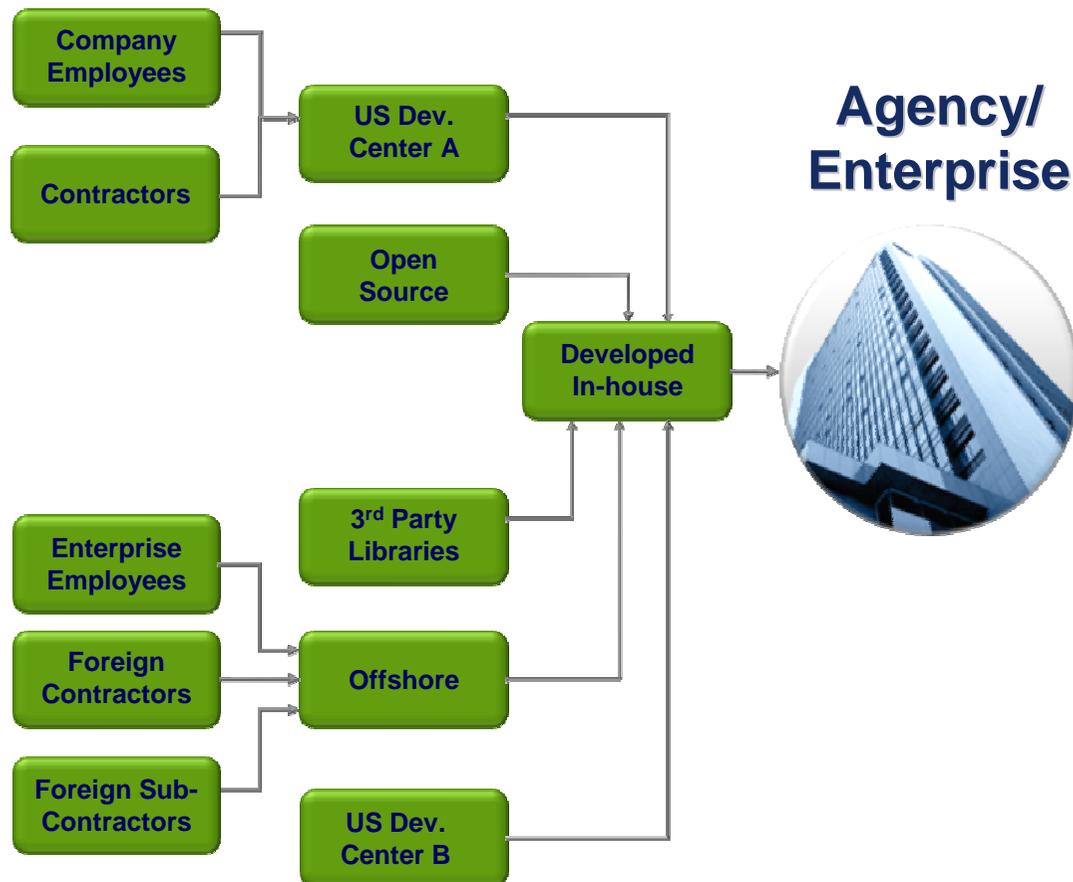
Focuses on the role of contractors in security as Federal agencies outsource various IT functions.



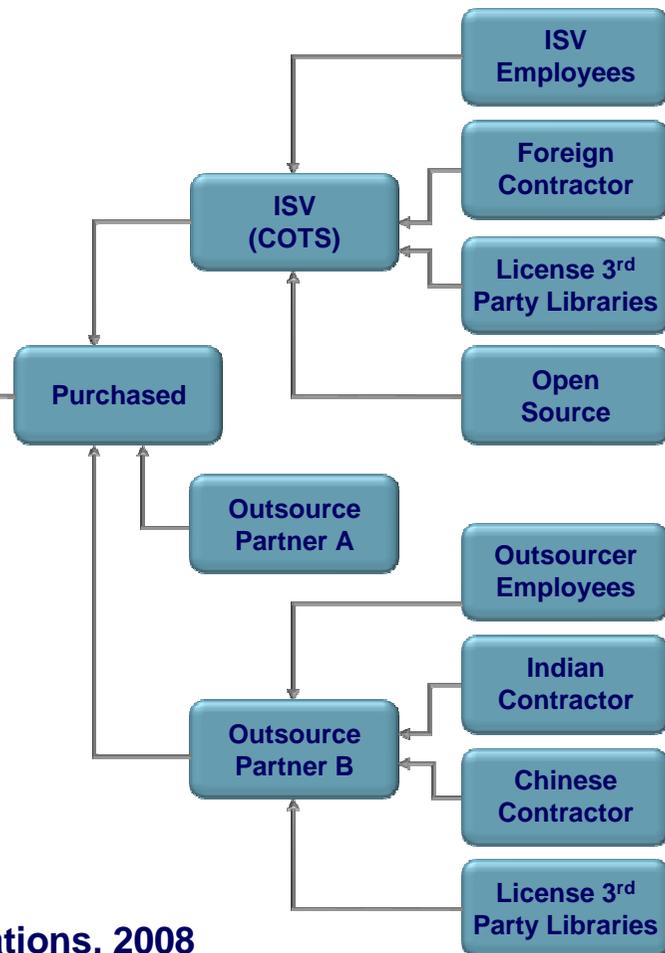
# Enterprise Processes for deploying capabilities: Increasingly Distributed and Complex

## New Considerations for Quality & Security

### Development Process



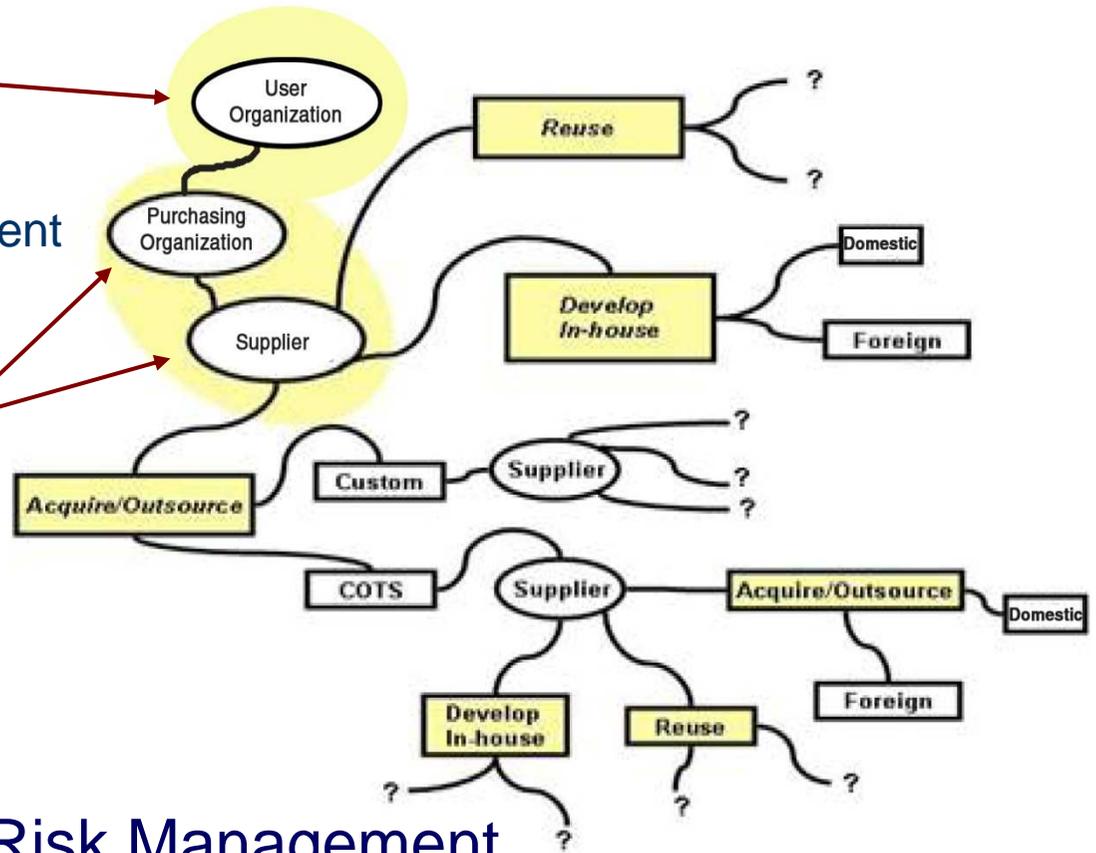
### Procurement Process



Source: SwA WG Panel presentations, 2008

# Risk Management (Enterprise $\Leftrightarrow$ 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



# Software Assurance ‘End State’ Objectives...

---

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



# Need for Rating Schemes

---



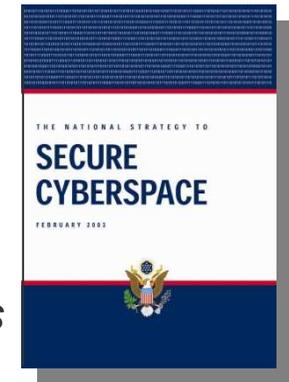
- ▶ Rating of Software products:
  - Supported by automation
  - Standards-based
  - Rules for aggregation and scaling
  - Verifiable by independent third parties
  - Labeling to support various needs (eg., security, dependability, etc)
  - Meaningful and economical for consumers and suppliers
  
- ▶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



# DHS Software Assurance Program Overview

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

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



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

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".

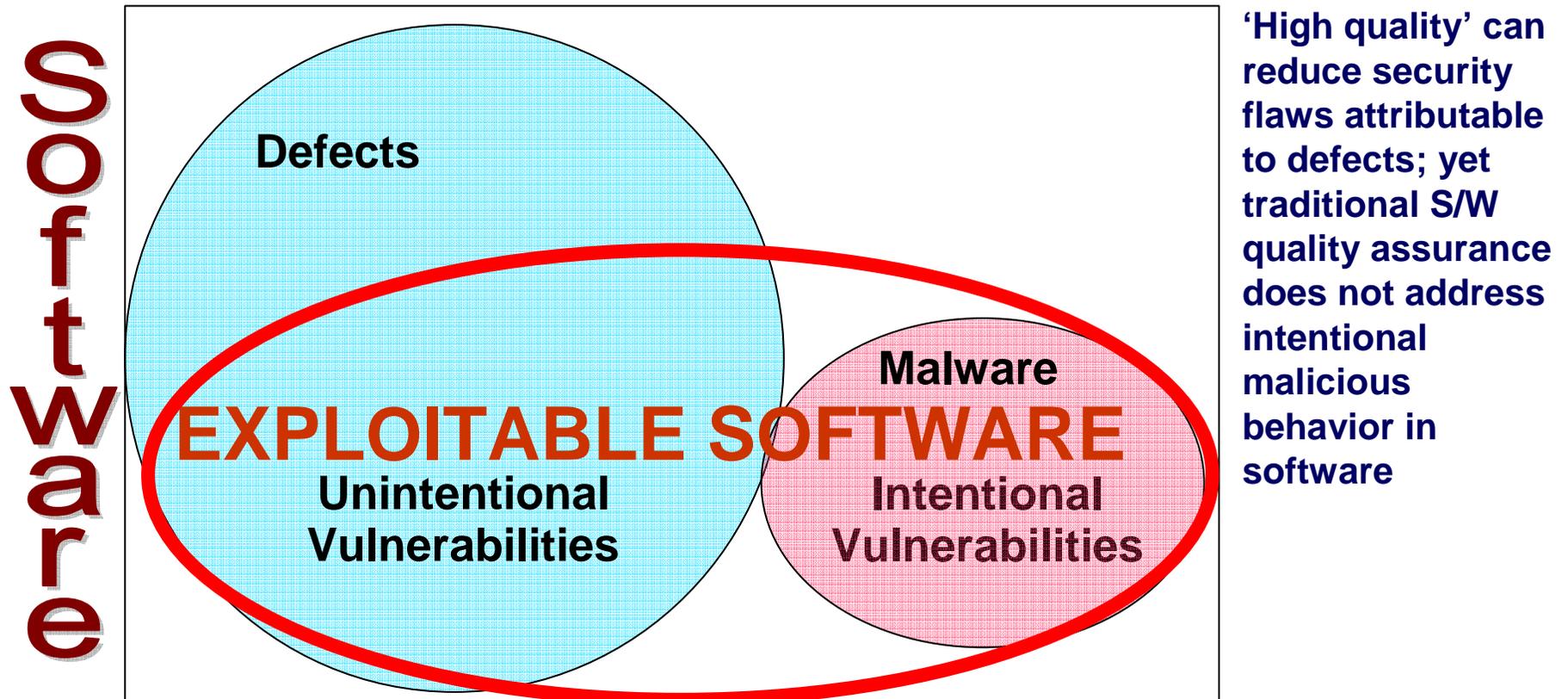


**Homeland  
Security**

# Software Assurance Addresses Exploitable Software:

Outcomes of non-secure practices and/or malicious intent

Exploitation potential of vulnerability is independent of “intent”



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



# DHS Software Assurance Program Structure \*

---

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





# Software Assurance Forum & Working Groups\*

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

| People   | Processes  | Technology  | Acquisition  |
|--|--|---|--|
| Developers and users education & training  | Sound practices, standards, & practical guidelines for secure software development | Security test criteria, diagnostic tools, common enumerations, SwA R&D, and SwA measurement   | Software security improvements through due-diligence questions, specs and guidelines for acquisitions/ outsourcing |
| <b>Products and Contributions</b>  |  |   |  |
| Build Security In - <a href="https://buildsecurityin.us-cert.gov">https://buildsecurityin.us-cert.gov</a> and SwA community resources & info clearinghouse<br><br>SwA Common Body of Knowledge (CBK) & Glossary<br>Organization of SwSys Security Principles/Guidelines<br>SwA Developers' Guide on Security-Enhancing SDLC<br><br>Software Security Assurance State of the Art Report<br>Systems Assurance Guide (via DoD and NDIA)<br><br>SwA-related standards – ISO/IEC JTC1 SC7/27/22, IEEE CS, OMG, TOG, & CMM-based Assurance |  | Practical Measurement Framework for SwA/InfoSec<br>Making the Business Case for Software Assurance<br><br>SwA Metrics & Tool Evaluation (with NIST)<br>SwA Ecosystem w/ DoD, NSA, NIST, OMG & TOG<br>NIST Special Pub 500 Series on SwA Tools<br><br>Common Weakness Enumeration (CWE) dictionary<br>Common Attack Pattern Enumeration (CAPEC)<br><br>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.

# 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.

# Software Assurance (SwA) Pocket Guide Series

## SwA in Acquisition & Outsourcing

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

## SwA in Development

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

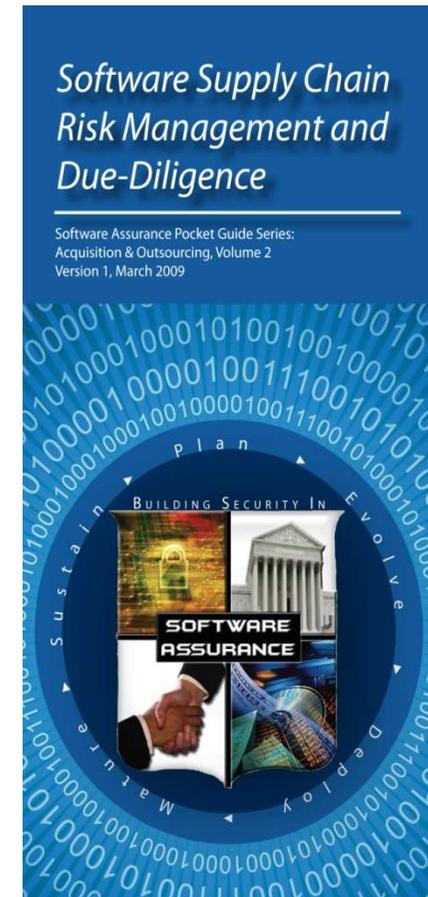
## SwA Life Cycle Support

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

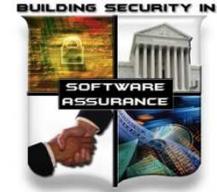
## SwA Measurement and Information Needs

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

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

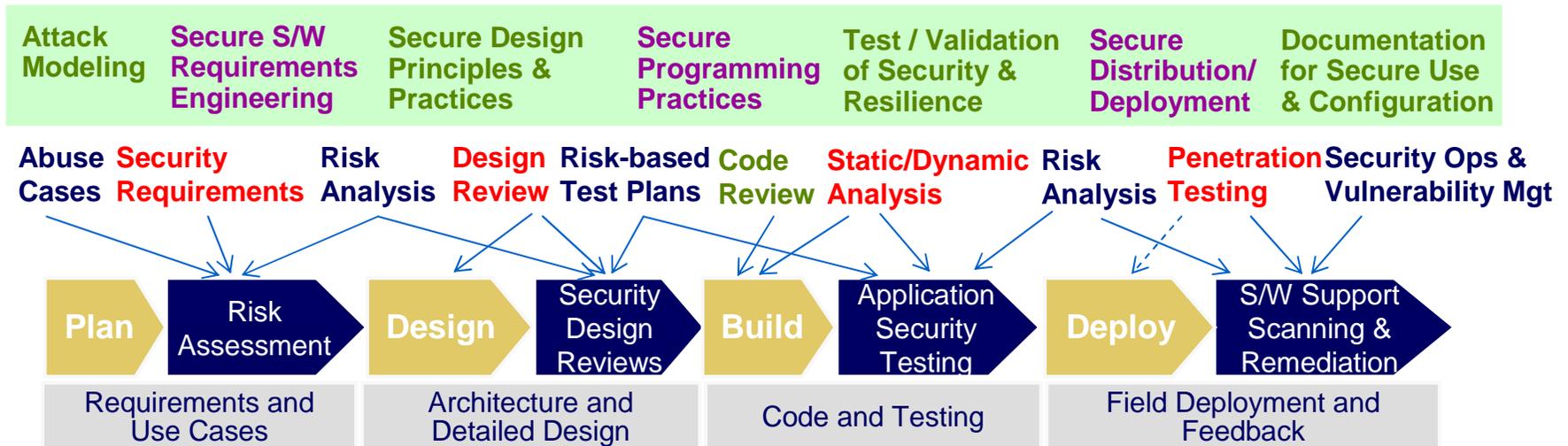


# Security-Enhanced Process Improvements



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

“Build Security In” throughout the lifecycle



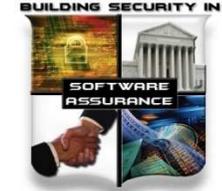
Organizational Process Assets cover: governance, policies, standards, training, tailoring guidelines

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



**Homeland Security**

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



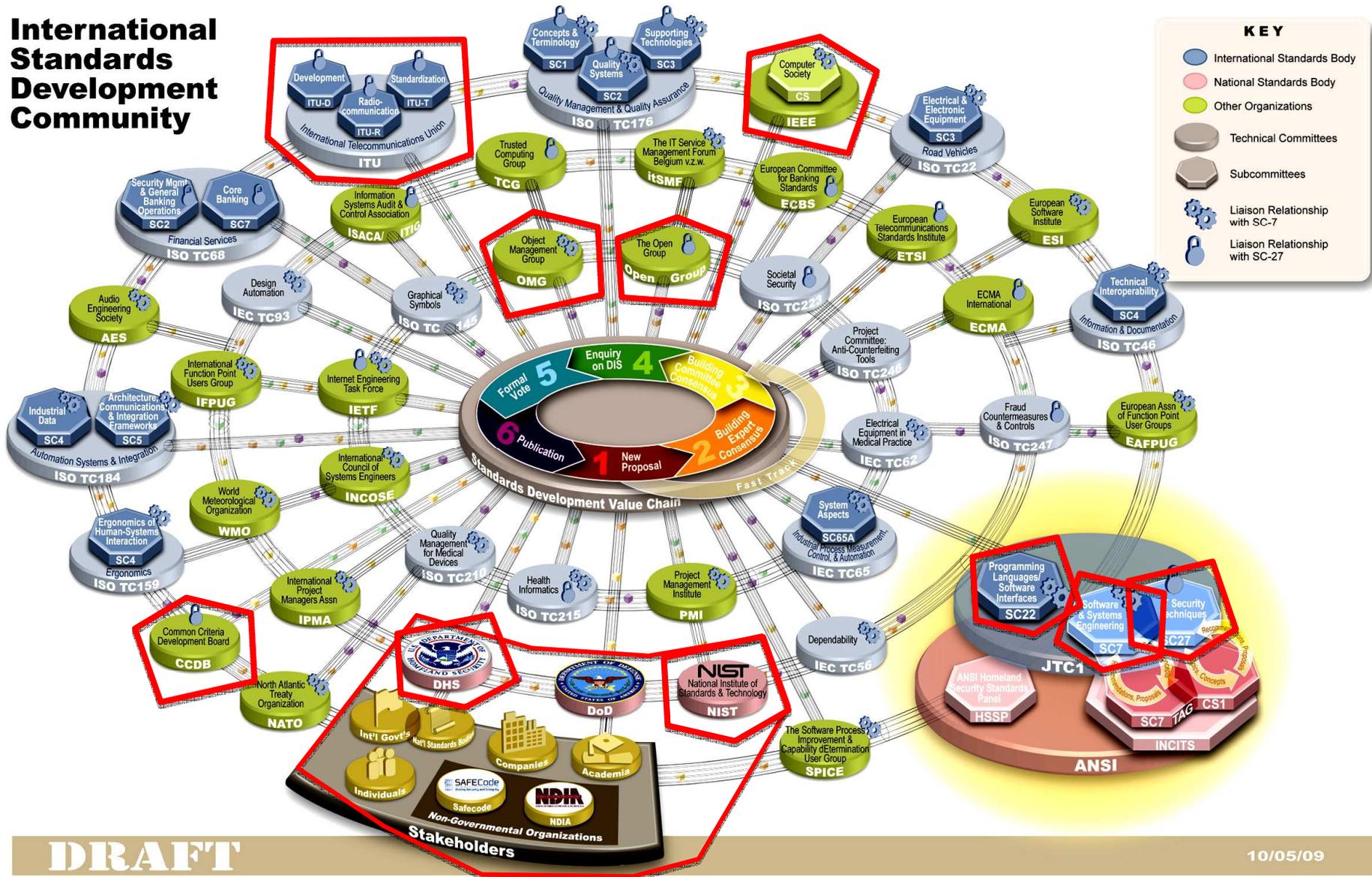
# Build Security In the SDLC

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



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

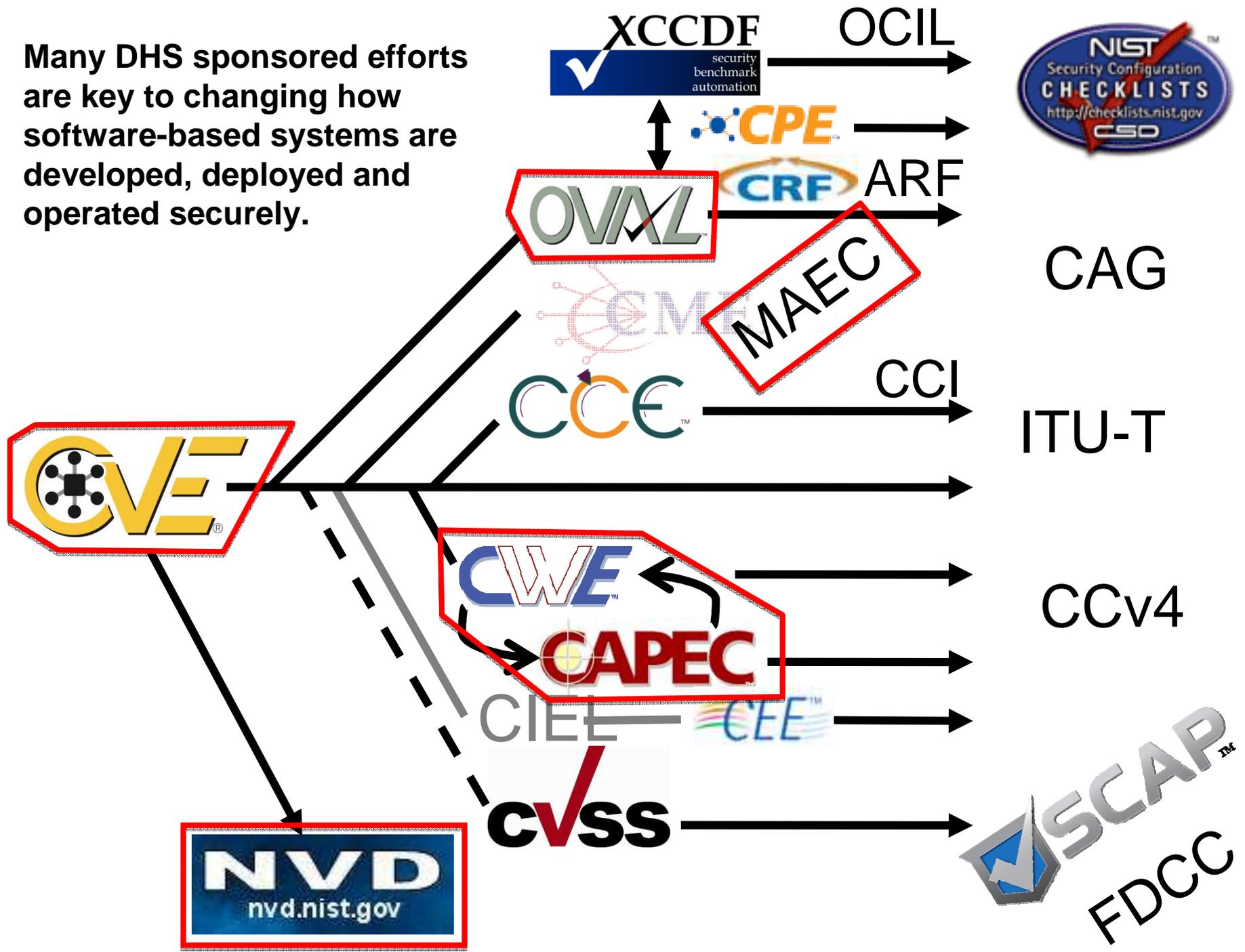
## International Standards Development Community



**DRAFT**

10/05/09

Many DHS sponsored efforts are key to changing how software-based systems are developed, deployed and operated securely.



# Industry, Government, and Academia



CAPEC

MAEC

CWE

CVSS

ARF

OVAL

CPE

OCI

NVD  
nvd.nist.gov

CVE

XCCDF  
security benchmark automation

CEE

NIST  
Security Configuration  
CHECKLISTS  
http://checklists.nist.gov  
CSO

FDCC

SCAP  
EMAP  
SwAAP



## NIST Special Publications:

|           |                                  |
|-----------|----------------------------------|
| SP800-36  | CVE                              |
| SP800-40  | CVE, OVAL                        |
| SP800-42  | CVE                              |
| SP800-44  | CVE                              |
| SP800-51  | CVE                              |
| SP800-53a | CVE, OVAL, CWE                   |
| SP800-61  | CVE, OVAL                        |
| SP800-70  | CVE, OVAL, CCE, CPE, XCCDF, CVSS |
| SP800-82  | CVE                              |
| SP800-86  | CVE                              |
| SP800-94  | CVE                              |
| SP800-115 | CVE, CCE, CVSS, CWE              |
| SP800-117 | CVE, OVAL, CCE, CPE, XCCDF, CVSS |
| SP800-126 | CVE, OVAL, CCE, CPE, XCCDF, CVSS |



**FDCC**

## NIST Interagency Reports:

|             |                                  |
|-------------|----------------------------------|
| NISTIR-7007 | CVE                              |
| NISTIR-7275 | CVE, OVAL, CCE, CPE, XCCDF, CVSS |
| NISTIR-7435 | CVE, CVSS, CWE                   |
| NISTIR-7511 | CVE, OVAL, CCE, CPE, XCCDF, CVSS |
| NISTIR-7517 | CVE                              |
| NISTIR-7581 | CVE                              |
| NISTIR-7628 | CVE, CWE                         |





**CWE**  
**Validation**  
 Effectiveness  
 Testing - ?

**CWE**  
 Compatibility  
 and  
 Effectiveness  
  
 CWEs with  
 WhiteBox  
 Definitions

**Center For**  
**Assure SW**  
 Tool Evaluation  
 2007  
 Tool Evaluation  
 2009  
  
**IARPA**  
 STONESOUP-  
 Securely Taking  
 On New  
 Executable Stuff  
 Of Uncertain  
 Provenance

**OSD/NII**  
 CWE  
 Formalization

**NIST**  
**SAMATE**  
 SP 500-267  
 SP 500-269  
 SP 500-270  
  
 SAMATE  
 Repository  
 Dataset  
 (SRD)  
  
 Automated  
 Test Case  
 Generator

**NIST SATE**  
**SATE08**  
**SATE09**

**SySA Task**  
**Force**  
 WhiteBox  
 Definitions-to-  
 SBVR-to-  
 microKDM

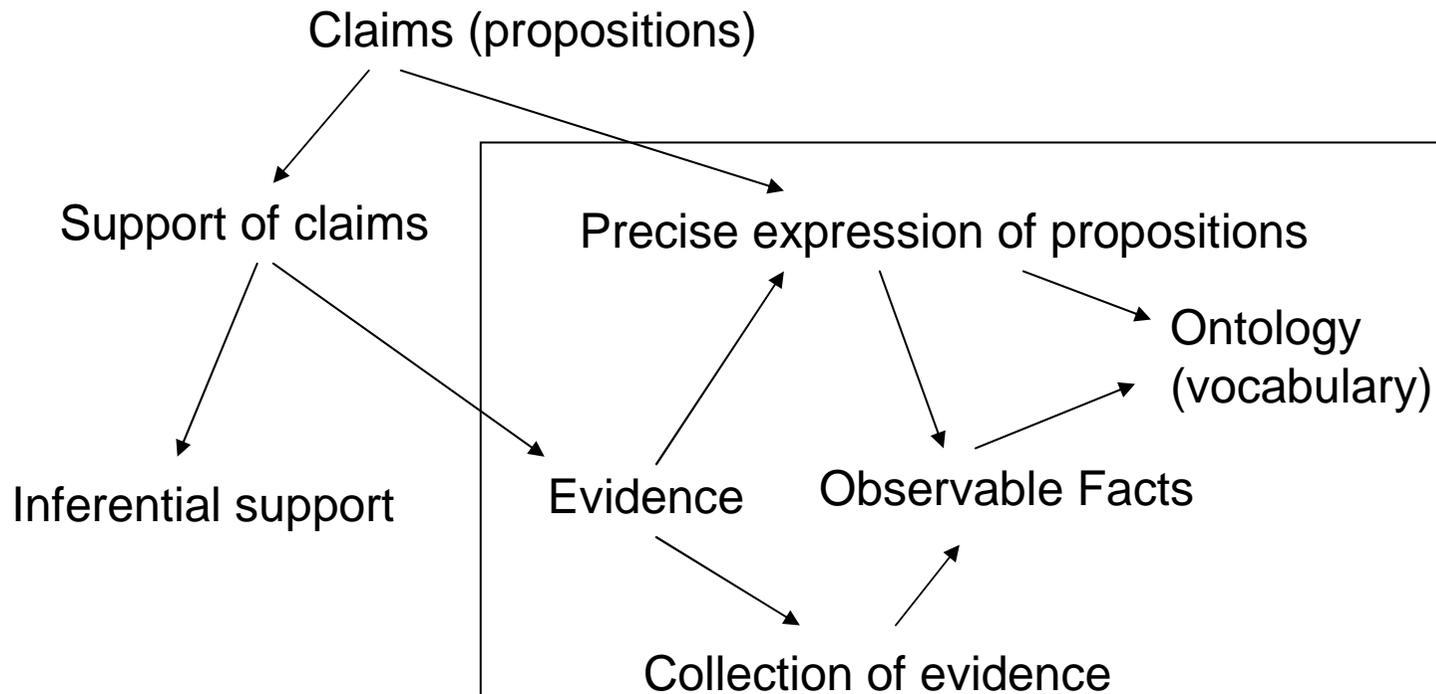
All of these are aimed at different aspects of understanding how well tools find CWEs in software applications and what can be done to improve that and standardize the process for expressing a tools capabilities.



# OMG Systems Assurance Task Force Claims-Evidence-Arguments Overview

Assurance Case

ARM Argumentation Metamodel



SBVR  
Semantic  
Business  
Vocabulary  
& Rules

SAEM Software Assurance Evidence Metamodel



KDM Analytics

KDM Knowledge Discovery Metamodel





ISO IEC

ISO/IEC JTC 1/SC 27 NXXXX

ISO/IEC JTC 1/SC 27/WG 3 NXXXXX

REPLACES: N

ISO/IEC JTC 1/SC 27

Information technology - Security techniques

Secretariat: DIN, Germany

DOC TYPE: NB MWI Proposal for a technical report (TR)

TITLE: National Body New Work Item Proposal on "Secure software development and evaluation under ISO/IEC 15406 and ISO/IEC 18405"

SOURCE: INCITSACS 1, National Body of (US)

DATE: 2006-09-30

PROJECT: 15406 and 18405

STATUS: This document is circulated for consideration at the forthcoming meeting of SC 27/WG 3 to be held in Redmond (WA, USA) on 2<sup>nd</sup> - 6<sup>th</sup> November 2006.

ACTION ID: ACT

DUE DATE:

DISTRIBUTION: P, D and L-Members  
W: Larry S.D. Z. Chairman  
M: De Smitte, SC 27 Vice-Chair  
E: J. Humphreys, K. Naraenara, Y. Buitón, M.-C. Kang, K. Ramming, WG-Consensus

MEDIUM: Live Intranet

NO. OF PAGES: xx

Secretariat ISO/IEC JTC 1/SC 27 -  
DIN Document Information for Germany e. V., Burgstraße 6, 10772 Berlin, Germany  
Telephone: +49 30 201-15552; Facsimile: +49 30 2501-7233; E-mail: [iso@iae.din.de](mailto:iso@iae.din.de)  
[HTTP://www.iso-iae.din.de/](http://www.iso-iae.din.de/)

## Common Criteria v4 CCDB

- TOE to leverage CAPEC & CWE
- Also investigating how to leverage ISO/IEC 15026

## NIAP Evaluation Scheme

- Above plus
- Also investigating how to leverage SCAP

### New Work Item Proposal

NP submitting

### PROPOSAL FOR A NEW WORK ITEM

|  |   |
|--|---|
| Date of presentation of proposal<br>YYYY-MM-DD | Proposer: ISO/IEC JTC 1/SC 27                 |
| Secretariat<br>National Body                   | ISO/IEC JTC 1 N XXXX<br>ISO/IEC JTC 1/SC 27 N |

A proposal for a new work item shall be submitted to the secretariat of the ISO/IEC joint technical committee concerned with a copy to the ISO Central Secretariat.

### Presentation of the proposal

Title: Secure software development and evaluation under ISO/IEC 15406 and ISO/IEC 18405

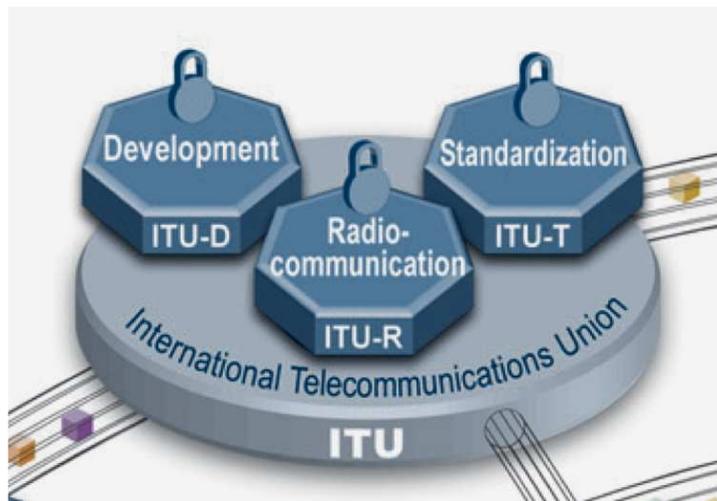
### Scope

In the case where a target of evaluation (TOE) being evaluated, under ISO/IEC 15406 and ISO/IEC 18405, includes specific software portions, the TOE developer may optionally present the developer's technical rationale for mitigating software common attack patterns and related weaknesses as described in the latest revision of the Common Attack Pattern Enumeration and Classification (CAPEC) model table from <http://capec.mitre.org/>. The developer's technical rationale is expected to include a range of mitigation techniques, from architectural properties to design features, coding techniques, use of tools or other means.

This Technical Report (TR) provides guidance for the developer and the evaluator on how to use the CAPEC as a technical reference point during the TOE development phase and in an evaluation of the TOE secure software under ISO/IEC 15406 and 18405, by addressing:

- A refinement of the IS 15406 Attack Potential evaluation table for software, taking into account the entries contained in the CAPEC and their characteristic.
- How the information for mitigating software common attack patterns and related weaknesses is used in an IS 15406 evaluation, in particular providing guidance on how to determine which attack patterns and weaknesses are applicable to the TOE, taking into consideration of:
  - the TOE technology;
  - the TOE security problem definition;
  - the interfaces the TOE exports that can be used by potential attackers;
  - the Attack Potentials that the TOE needs to provide resistance for.
- How the technical rationale provided by the developer for mitigating software common attack patterns and related weaknesses is used in the evaluation of the TOE design and the development of test cases.
- How the CAPEC and related Common Weakness Enumeration (CWE) weaknesses are used by the evaluator, who needs to consider at the applicant attack patterns and be able to exploit specific related software weaknesses while performing the subsequent vulnerability analysis (AVA\_VAN) activities on the TOE.
- How incomplete entries from the CAPEC are resolved during an IS 15406 evaluation.
- How the evaluator's attack and weakness analysis of the TOE incorporates other attacks and weaknesses not yet documented in the CAPEC.

The TR also investigates specific elements from the ISO/IEC 15026 (and its revision) are applicable to the guidelines being developed in the TR within the context of IS 15406 and 18405.

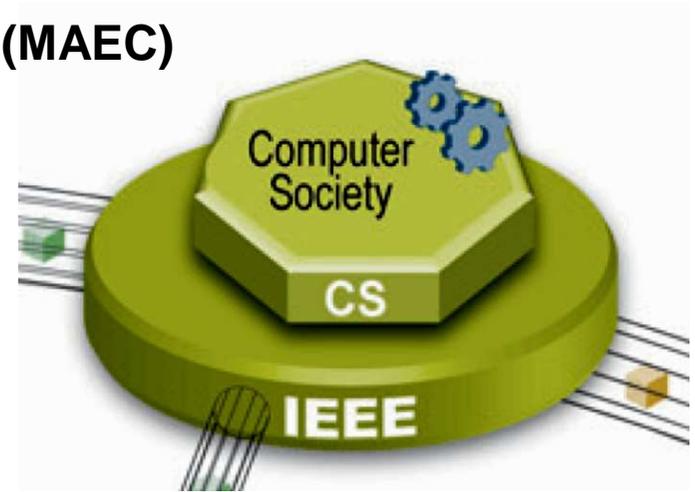
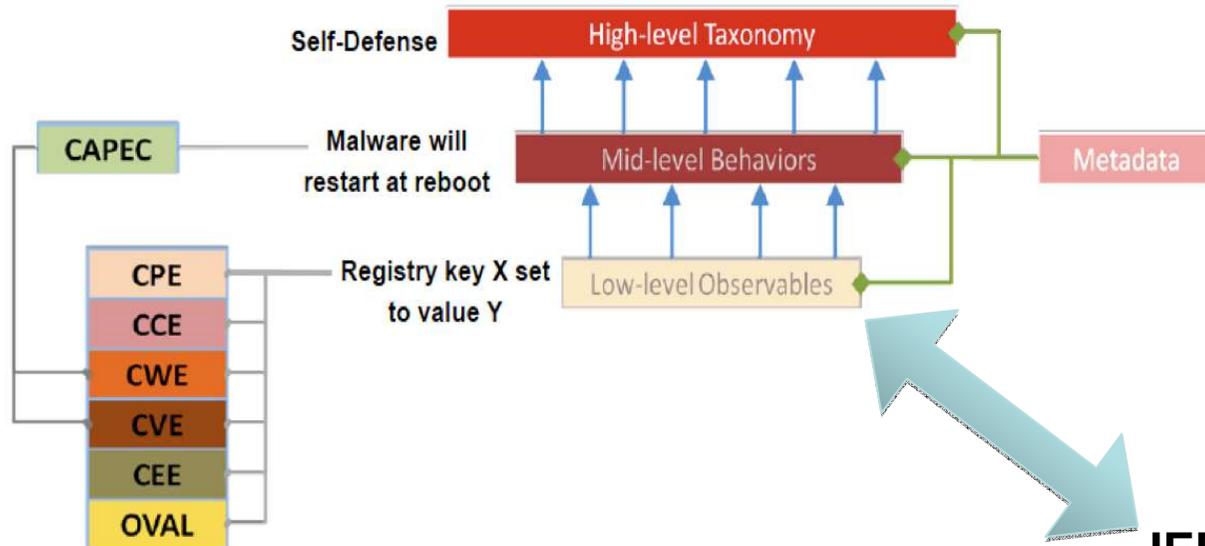


## ITU-T Study Group 17 Question 4 – Cyber Security Cyber Security Exchange Framework (CYBEX)

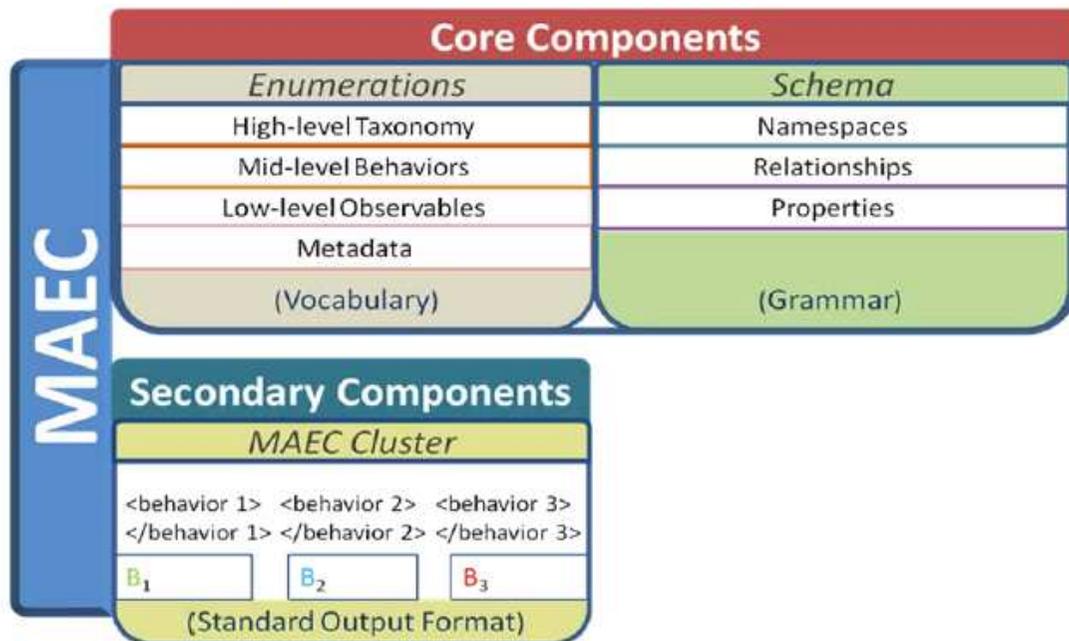
Creating x.series standards to capture the correct and supported USE of the enumerated concepts and languages – effort stewardship and definition stays with originating organizations

| <u>Identifier</u>     | <u>Title</u>  | <u>Current Text</u> |
|-----------------------|---|---------------------|
| X.cybief              | Cybersecurity Information Exchange Framework                                    | TD406               |
| X.cybief.1            | Guidelines for Administering the OID arc for cybersecurity information exchange | TD406               |
| <b>X.cce</b>          | <b>Common Configuration Enumeration</b>   | TD406               |
| <b>X.cee</b>          | <b>Common Event Expression</b>  | TD406               |
| X.chirp               | Cybersecurity Heuristics and Information Request Protocol                       | TD406               |
| <b>X.cpe</b>          | <b>Common Platform Enumeration</b>  | TD406               |
| <b>X.crf</b>          | <b>Common Result Format</b>   | TD406               |
| <b>X.cve</b>          | <b>Common Vulnerabilities and Exposures</b>                                     | TD405               |
| <b>X.cvss</b>         | <b>Common vulnerability scoring system</b>                                      | TD412               |
| <b>X.cwe</b>          | <b>Common Weakness Enumeration</b>  | TD406               |
| <b>X.cwss</b>         | <b>Common Weakness Scoring System</b>   | TD406               |
| X.dexf                | Digital evidence exchange file format   | C97                 |
| X.dpi                 | Deep Packet Inspection Exchange Format  | TD406               |
| X.gridf               | SmartGrid Incident Exchange Format  | TD406               |
| <b>X.oval</b>         | <b>Open Vulnerability and Assessment Language</b>                               | TD406               |
| X.pfoc                | Phishing, Fraud, and Other Crimeware Exchange Format                            | TD406               |
| <b>X.scap</b>         | <b>Security Content Automation Protocol</b>                                     | TD406               |
| X.teef                | Cyber attack tracing event exchange format                                      | C135, C129          |
| <b>X.xccdf</b>        | <b>eXensible Configuration Checklist Description Format</b>                     | TD406               |
| X.cybief-[namespace], | Cybersecurity Information Exchange Namespace                                    | C148                |
| X.cybief-discovery    | Cybersecurity Information Exchange Discovery                                    | C145                |
| <b>X.capec</b>        | <b>Common Attack Pattern Enumeration and Classification</b>                     | TD406               |
| X.iodef               | Incident Object Description Exchange Format                                     | TD406               |

# Malware Attribute Enumeration and Characterization (MAEC)



## MAEC High-level Overview

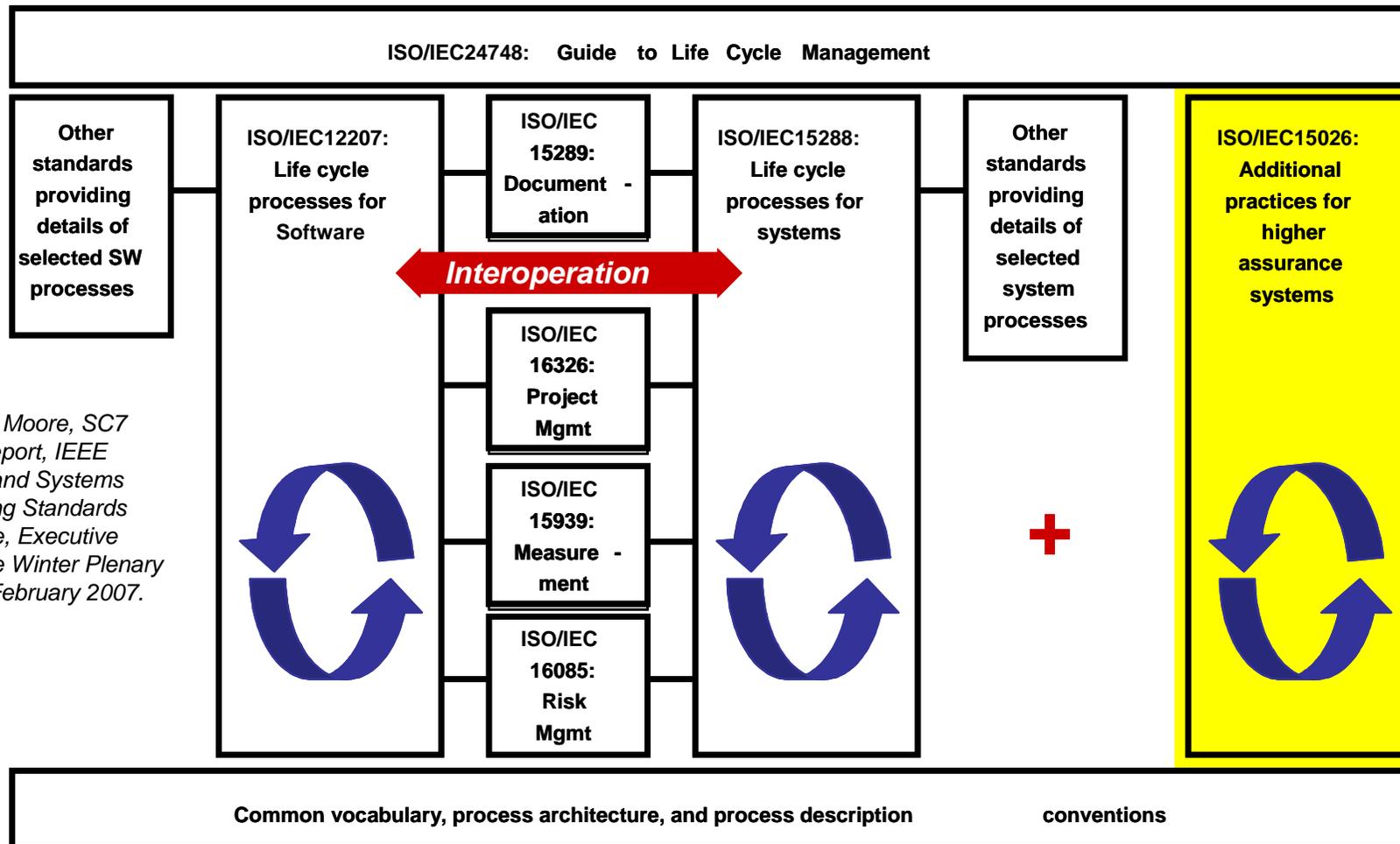


## IEEE's Industry Connections Security Group (ICSG)

First working group is focused on malware (malicious software such as viruses, worms and spyware).

Microsoft, McAfee, Symantec, Sophos, AVG, and Trend

# ISO/IEC/IEEE 15026, System and Software Assurance



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

“System and software assurance focuses on the management of risk and assurance of safety, security, and dependability within the context of system and software life cycle  
 Terms of Reference changed: ISO/IEC JTC1/SC7 WG7, previously “System and Software Integrity” SC7 WG9

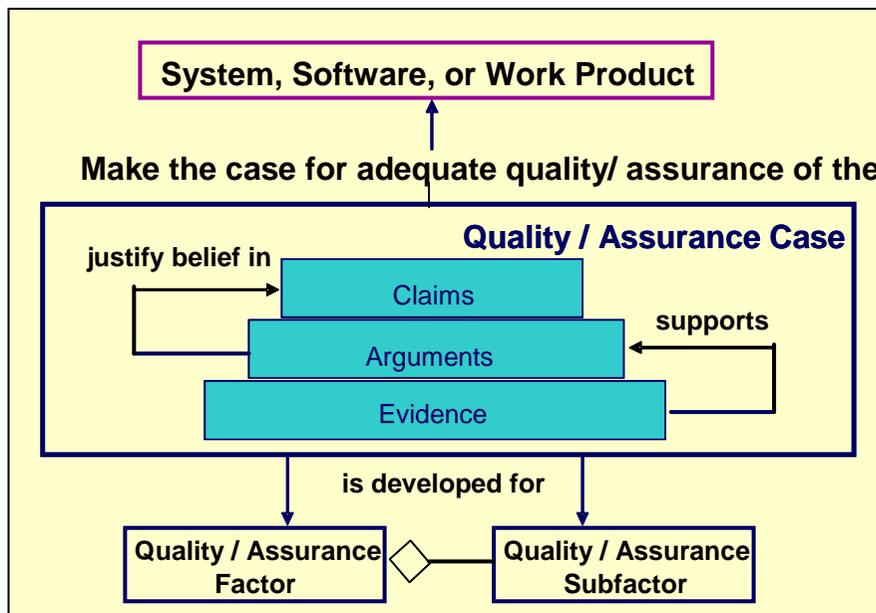
# ISO/IEC/IEEE 15026 Assurance Case

- **Set of structured assurance claims, supported by evidence and reasoning (arguments), that demonstrates how assurance needs have been satisfied.**

- Shows compliance with assurance objectives
- Provides an argument for the safety and security of the product or service.
- Built, collected, and maintained throughout the life cycle
- Derived from multiple sources

- **Sub-parts**

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



## Attributes

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

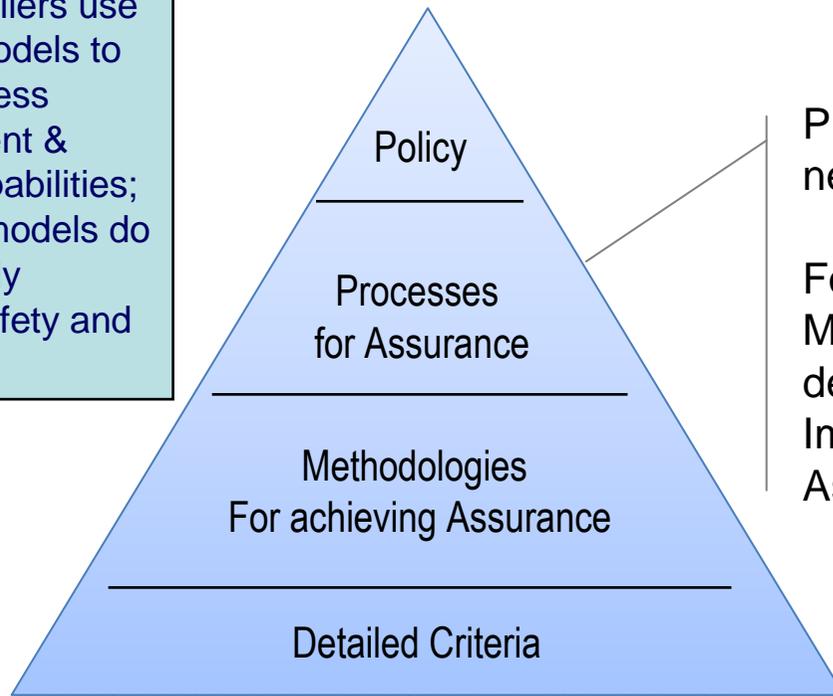
# The Landscape of Cyber Security Standardization Efforts

|   | Standard Processes   |  | Standard Formats & Concepts  |  | Common Collections/Reference Resources |   |
|---|--|--|--|--|--|---|
|   | IT   | Cyber Security   | IT   | Cyber Security   | IT                                     | Cyber Security  |
| <b>Pre-Deployment Phase</b>             | <p>24748: Guide to Life Cycle Management</p> <p>12207: Life cycle processes for SW</p> <p>16326: Project Mgmt</p> <p>15939: Measurement</p> <p>16085: Risk Management</p> <p>15288: Life cycle processes for systems</p> | <p>15026: Additional practices for higher assurance systems</p> <p>Common Criteria</p> | <p>ISO/IEC SC22 collection of language standards</p> <p>OMG KDM - Knowledge Discovery Metamodel</p> <p>OMG SBVR - Symantec Business Vocabulary and Rules</p> | <p>24772 PL vulnerabilities</p> <p>OMG SAEM –SW Assurance Evidence Metamodel</p> <p>OMG ARG – Argumentation Metamodel</p> <p>X.CWE</p> <p>X.CAPEC</p>                                      | SWEBOK                                 | <p>CWE</p> <p>CAPEC</p> <p>SWEBOK Security KA</p> <p>ISSA CCLSP Assurance-related questions</p> <p>SE2004 curriculum Curriculum proposals</p> <p>ABET accreditation</p> <p>CSDP Assurance-related questions</p> |
| <b>Post-Deployment Operations Phase</b> | ITIL   | <p>27000</p> <p>SP800-53 and 53a</p>   |  | <p>SP800-117</p> <p>SP800-126</p> <p>X.CVE</p> <p>X.CVSS</p> <p>X.OVAL</p> <p>X.XCCDF</p> <p>X.CCE</p> <p>X.CPE</p> <p>X.CWE</p> <p>X.CAPEC</p> <p>X.CEE</p> <p>X.MAEC</p> <p>X.CYBIEF</p> | <p>DNS</p> <p>GRC Roundtable</p>       | <p>FDCC</p> <p>SCAP</p> <p>NVD</p> <p>CVE</p> <p>CVSS</p> <p>OVAL</p> <p>XCCDF</p> <p>CCE</p> <p>CPE</p> <p>CWE</p> <p>CAPEC</p> <p>CEE</p> <p>MAEC</p>   |



# 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)<sup>®</sup> 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/procesrc.html>

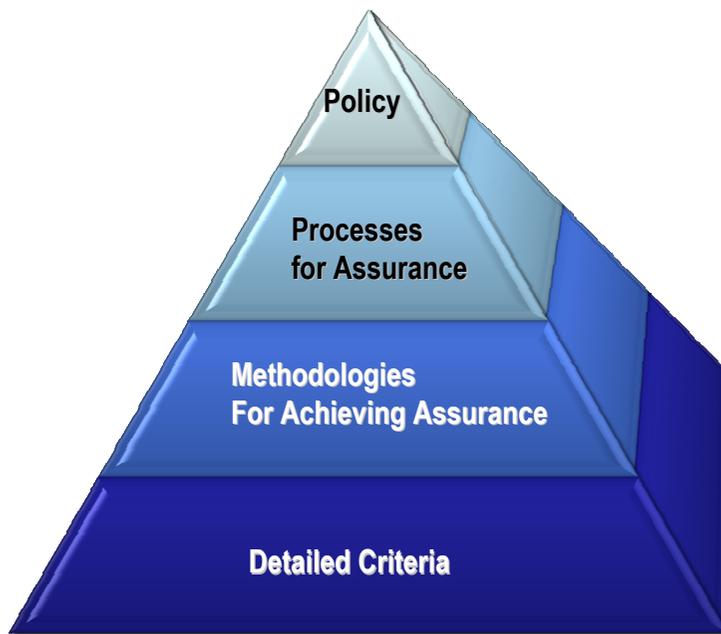
Experience gained for “Assurance” enhanced processes in *U.S. DoD and FAA joint project on Safety and Security Extensions for Integrated Capability Maturity Models, September 2004*, at SwA Community Resources and Information Clearinghouse - see <https://buildsecurityin.us-cert.gov/swa/downloads/SafetyandSecurityExt-Sep2004.pdf>

## **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/>



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



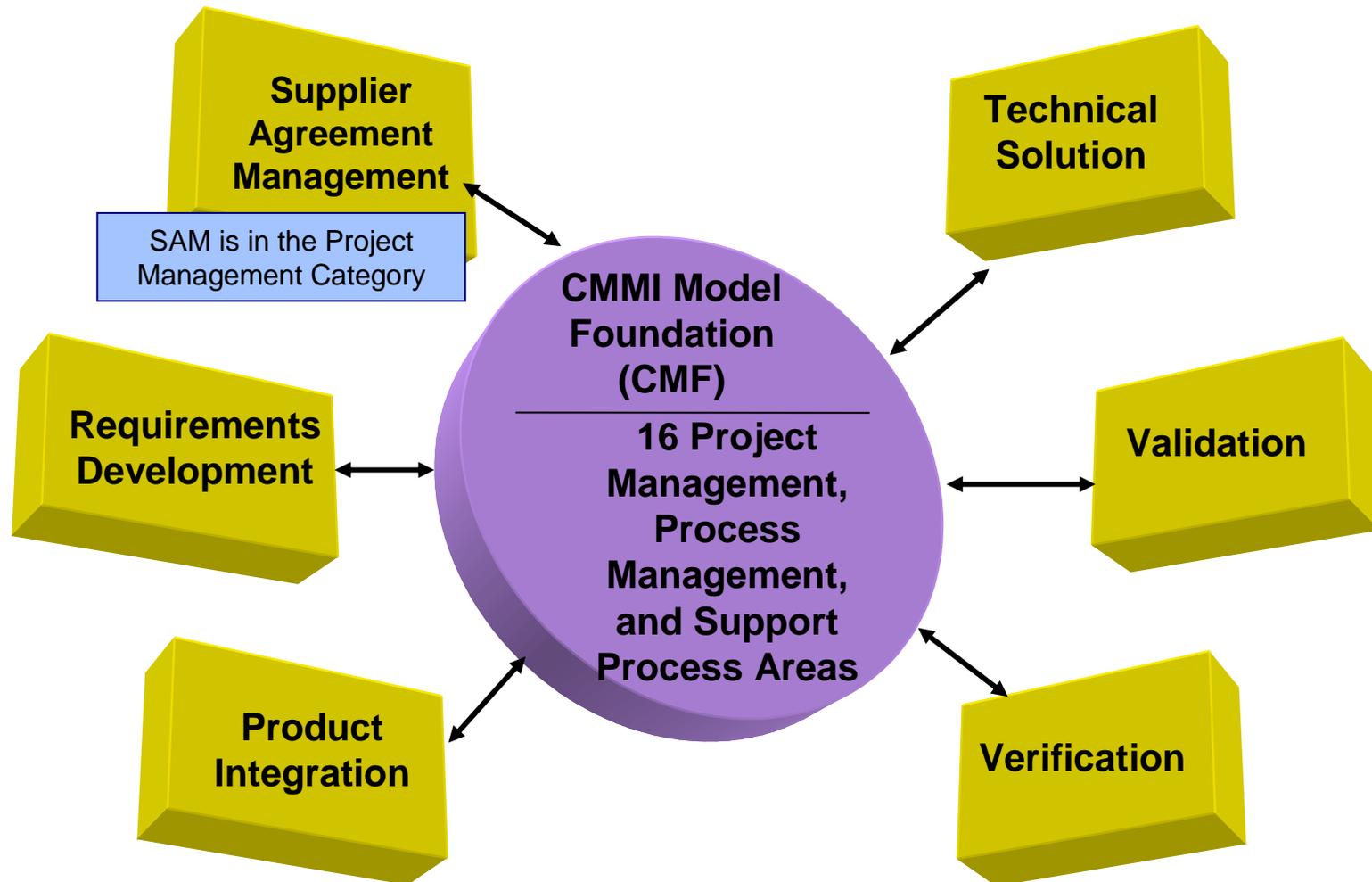
- Assurance PRM defines the goals and practices needed to achieve SwA
- Assurance for CMMI ® defines the Assurance Thread for Implementation and Improvement of Assurance Practices that are assumed when using the CMMI-DEV



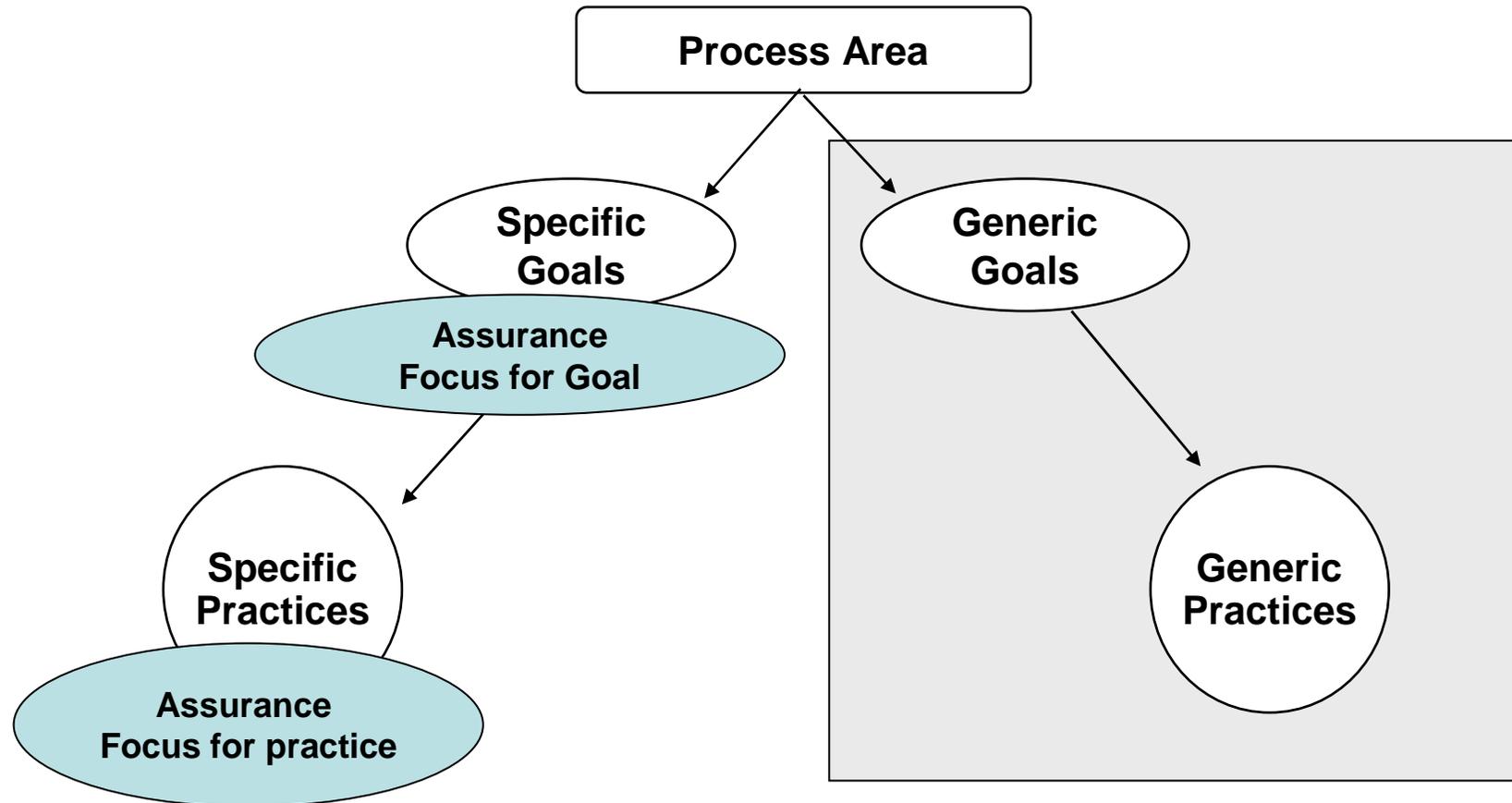
Understanding gaps helps suppliers and acquirers prioritize organizational efforts and funding to implement improvement actions

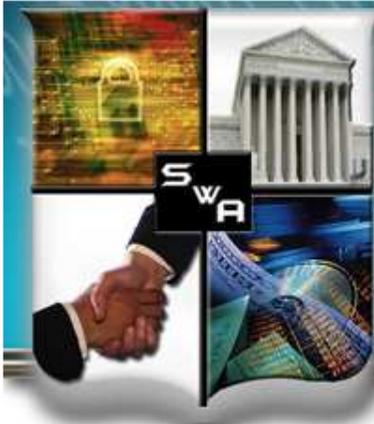
<https://buildsecurityin.us-cert.gov/swa/procesrc.html>

*Assurance for Capability Maturity Model Integration  
(CMMI)<sup>®</sup> -- CMMI-DEV v1.2*



# ***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



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



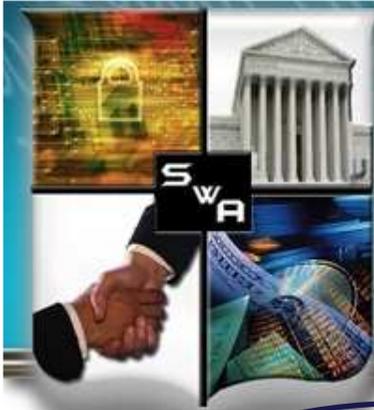
Homeland  
Security



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



Homeland  
Security

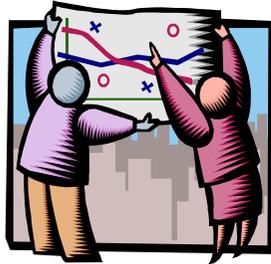


# SOFTWARE ASSURANCE FORUM

## BUILDING SECURITY IN Process Improvement Lifecycle - A Process for Achieving Assurance

### Mission/Business Process

Understand Your Business Requirements for Assurance



### Measure Your Results



### Information System

Build or Refine and Execute Your Assurance Processes



Understand Assurance-Related Process Capability Expectations



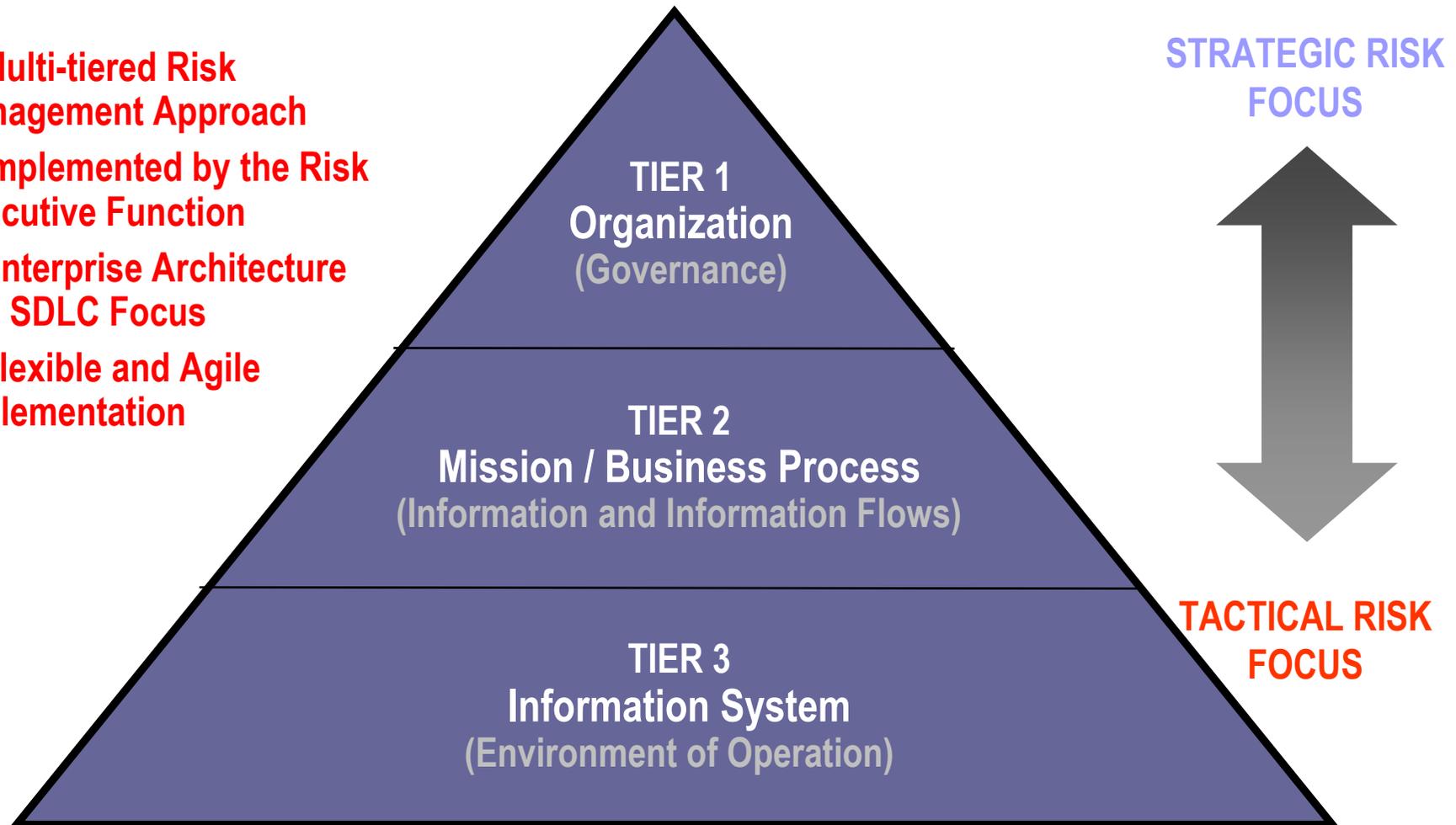
### Organization Support

Look to Standards for Assurance Process Detail



# Enterprise-Wide Risk Management

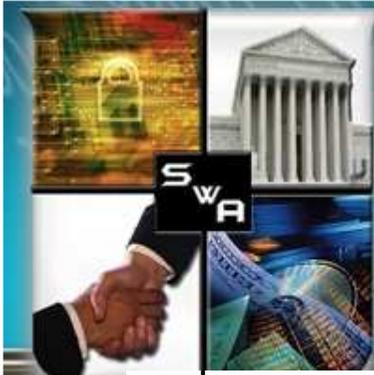
- Multi-tiered Risk Management Approach
- Implemented by the Risk Executive Function
- Enterprise Architecture and SDLC Focus
- Flexible and Agile Implementation



*FISMA 2010 and Beyond*  
*Strategic and Tactical Risk Management and the Role of Software Assurance*  
*Ron Ross, NIST*  
*Software Assurance Workshops*  
*June 21, 2010*



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



# SOFTWARE ASSURANCE FORUM

## BUILDING SECURITY IN

### Mappings Of The Common Practices

SWA Common Practices Consolidation

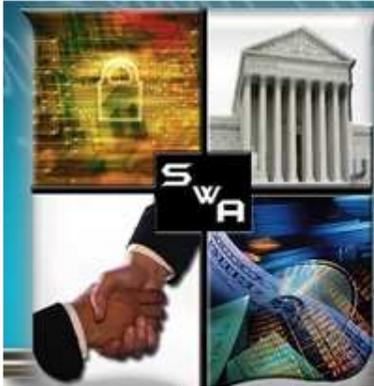
|                   | Governance   |   |  | Knowledge   |   |   | Verification  |  |   | Deployment  |   |  | Supplier Management   |  |   |          |
|-------------------|--|---|--|---|---|---|---|--|---|---|---|--|---|--|---|----------|
|                   | Strategy & Metrics   | Policy & Compliance   | Training & Guidance  | Threat Assessment   | Security Requirements   | Secure Design   | Architecture Analysis                                       | Code Analysis  | Risk-Based Security Testing   | Penetration Testing   | Vulnerability Management  | Environment Hardening  | Agreement Requirements  | Evaluation & Selection                                     | Agreement Management  |          |
| <b>Practices:</b> | Establishes Security Plan; communicates and provides training for the plan | Identifies and monitors relevant compliance drivers   | Conducts security awareness training regularly             | Builds and maintains list of application-specific attack models | Documents, analyzes, and manages functional security requirements             | Develops list of preferred frameworks and security features; explicitly applies security principles to design | Reviews design against security requirements                | Develops list of top bugs and creates review checklists from security requirements | Performs edge boundary value condition testing in QA process                | Performs external penetration testing on production software with latest techniques and mitigates | Identifies point of contact for incident response; creates incident response team | Maintains operational environment specification  | Identifies and prioritizes supplier dependencies; identifies, assesses, and mitigates risks associated with supplier dependencies | Establishes, reviews, and distributes solicitation package | Formalizes supplier relationships and executes supplier agreement |          |
| <b>BSIMM</b>      | SM1.1  | CP1.1   | T1.1   | AM1.1   | SR1.1   | SFD1.1  | AA1.1 - AA1.3   | CR1.1  | ST1.1 - ST1.2   | PT1.1 - PT1.2   | CMVM2.1   | SE1.1  | SR3.1   | -  | -   |          |
| <b>CMMI-ACQ</b>   | PP SG2 - SG3   | OPF SG1   | OT SG2   | RSKM SG1 - SG2  | ARD SG1, SG3  | ATM SG2   | REQM SG1  | AVAL SG2   | AVAL SG1 - SG2  | AVER SG3  | AVER SG3  | CAR SG1  | CM SG2 - SG3  | RSKM SG2-SG3   | SSAD SG1  | AM SG1   |
| <b>OSAMM</b>      | SM1B   | PC1A  | EG1A   | TA1A  | SR1A  | SA1A  | DR1B  | CR1A   | ST1B  | ST1B  | VM1A  | EH1A   | -   | -  | -   | -        |
| <b>PRM</b>        | SG 2.1   | SG 3.1  | SG 1.3   | SG 3.2  | SG 3.1  | SG 3.2  | SG 3.4  | SG 3.4   | SG 3.4  | SG 3.4  | SG 4.3  | SG 4.3   | SG 2.3  | SG 2.3   | SG 2.3  | SG 2.3   |
| <b>PRM</b>        | SG 1.3   | -   | -  | -   | -   | -   | -   | -  | -   | -   | -   | -  | SG 3.1  | -  | -   | -        |
| <b>RMM</b>        | RTSE:SG2 - SG3   | COMP:SG2  | OTA:SG1 - SG2  | RISK:SG1 - SG4  | PRD:SG1 - SG3   | RTSE:SG1 - SG2  | -   | VAR:SG2  | RTSE:SG3  | RTSE:SG3  | VAR:SG1   | ADM:SG5  | EXD:SG1 - SG2   | EXD:SG3  | EXD:SG3   | EXD:SG3  |
| <b>RMM</b>        | MON:SG1  | MON:SG1 - SG2   | -  | KIM:SG6   | RRM:SG1   | KIM:SG2, SG6  | -   | KIM:SG2  | -   | -   | KIM:SG1   | KIM:SG5  | RISK:SG3 - SG6  | -  | -   | -        |
| <b>Practices:</b> | Collects and tracks security plan metrics based upon risk                  | Establishes policies and procedures for compliance with security plan and other compliance requirements | Conducts role-based advanced application security training | Identifies potential attacker profiles                          | Documents, analyzes, and manages non-functional security requirements         | Builds secure frameworks, security services, and security design patterns                                     | Makes design reviews available for projects                 | Uses automated code analysis tools; requires code analysis as part of development  | Integrates black box security testing tools into QA of software releases    | Performs periodic internal white box pen testing  | Develops consistent incident response process                                     | Monitors baseline environment configuration changes                                    | Establishes enterprise and assurance requirements for supplier agreement  | Evaluates solicitation responses                           | Monitors and corrects supplier processes and performance          |          |
| <b>BSIMM</b>      | SM1.5  | CP1.3   | T2.1   | AM1.3   | SR1.3   | SFD2.1  | AA2.1   | CR1.4  | ST2.1   | PT2.1 - PT2.3   | CMVM1.1   | SE1.1  | SR2.1, SR2.5  | -  | -   | -        |
| <b>BSIMM</b>      | SM2.1  | CP3.2   | -  | -   | -   | SFD2.3  | AA2.3   | CR2.3  | -   | -   | -   | -  | -   | -  | -   | -        |
| <b>CMMI-ACQ</b>   | MA SG1 - SG2   | OPF SG2 - SG3   | OT SG2   | RSKM SG1 - SG2  | ARD SG1, SG3  | ATM SG2   | AVAL SG1  | AVER SG3   | AVER SG3  | AVER SG3  | CAR SG1   | CM SG2 - SG3   | REQM SG1  | SSAD SG2   | AM SG1  | REQM SG1 |
| <b>CMMI-ACQ</b>   | PMC SG1  | -   | -  | -   | REQM SG1  | AVAL SG2  | PMC SG1 - SG2   | -  | -   | -   | OPD SG1   | -  | ARD SG2   | -  | REQM SG1  | -        |
| <b>OSAMM</b>      | SM1B   | PC2A  | EG2A   | TA1B  | SR1B  | SA2A  | DR2A  | CR2A   | ST1B  | ST1A  | VM2A  | EH2B   | SR3A  | -  | -   | -        |
| <b>OSAMM</b>      | -  | EG3B  | -  | -   | -   | SA2B  | DR2B  | CR2B   | ST1B  | ST1B  | -   | -  | -   | -  | -   | -        |
| <b>PRM</b>        | SG 1.1   | SG 1.2  | SG 1.3   | SG 3.2  | SG 3.1  | SG 3.2  | SG 3.4  | SG 3.4   | SG 3.4  | SG 3.4  | SG 4.3  | SG 4.3   | SG 3.1  | SG 2.3   | SG 2.3  | SG 3.5   |
| <b>PRM</b>        | SG 2.2   | -   | -  | -   | -   | -   | -   | -  | -   | -   | -   | -  | -   | -  | -   | -        |
| <b>RMM</b>        | MA:SG2   | RTSE:SG2  | OTA:SG3 - SG4  | RISK:SG1 - SG4  | COMP:SG2  | RTSE:SG3  | -   | RTSE:SG3   | RTSE:SG3  | RTSE:SG3  | VAR:SG1   | ADM:SG3  | EXD:SG3   | EXD:SG3  | EXD:SG3   | EXD:SG4  |
| <b>RMM</b>        | MON:SG2  | COMP:SG1  | -  | KIM:SG6   | RRM:SG1   | -   | -   | -  | -   | -   | MON:SG1   | KIM:SG5  | PRD:SG2 - SG3   | -  | RRM:SG1   | -        |
| <b>Practices:</b> | Drives budgets based upon analysis from metrics collections                | Measures project compliance at specific checkpoints   | Provides security resources for coaching / learning        | Builds and maintains abuse cases and attack patterns            | Builds repository of well written testable and reusable security requirements | Requires use of approved security platforms and architectures   | Builds standard architectural patterns from lessons learned | Tailors code analysis for application-specific concerns                            | Employs risk-driven automated security and regression testing in QA process | Performs extensive penetration testing customized with organizational knowledge                   | Conducts root cause analysis for incidents; fixes all occurrences of bugs         | Identifies and deploys relevant operations and protection tools; performs code signing | Establishes supplier agreement  | Negotiates and selects supplier                            | Evaluates and accepts supplier work products                      |          |
| <b>BSIMM</b>      | SM1.5  | CP2.3   | T1.3 - T1.4  | AM2.1   | SR1.2   | SFD3.2  | AA3.2   | CR3.1  | ST3.1   | PT3.1 - PT3.2   | CMVM3.1 - 3.2   | SE2.3  | CP2.4   | -  | -   | -        |
| <b>BSIMM</b>      | -  | CP3.3   | T2.4 - T2.5  | AM2.2   | SR2.3   | -   | -   | -  | -   | -   | -   | -  | CP3.2   | -  | -   | -        |
| <b>CMMI-ACQ</b>   | PMC SG2  | OPF SG1   | OT SG2   | RSKM SG2  | -   | CM SG1  | AVAL SG2  | AVER SG3   | AVER SG3  | AVER SG3  | CAR SG1 - SG2   | OID SG1 - SG2  | SSAD SG3  | SSAD SG2   | AM SG1  | PPQA SG1 |
| <b>CMMI-ACQ</b>   | -  | -   | -  | -   | -   | -   | -   | -  | -   | -   | -   | -  | -   | -  | -   | -        |
| <b>OSAMM</b>      | SM3A   | PC3A  | EG1B - EG2B  | TA2A  | SR2A  | SA3A  | DR3A  | CR3A   | ST1A  | ST1B  | VM3A  | EH3A   | -   | -  | -   | -        |
| <b>OSAMM</b>      | SM3B   | -   | EG3A   | -   | -   | SA3B  | -   | -  | ST2A  | -   | -   | OE3B   | -   | -  | -   | -        |
| <b>PRM</b>        | SG 3.1   | SG 4.1  | SG 1.3   | SG 3.1  | -   | SG 3.2  | SG 3.4  | SG 3.4   | SG 3.4  | SG 3.4  | SG 4.2  | SG 4.3   | SG 2.3  | SG 2.3   | SG 2.3  | SG 2.3   |
| <b>PRM</b>        | -  | -   | -  | -   | -   | -   | -   | -  | -   | -   | SG 3.5  | -  | -   | -  | -   | -        |
| <b>RMM</b>        | RTSE:SG3-SG1   | RTSE:SG2  | OTA:SG2  | RISK:SG1 - SG4  | KIM:SG6   | KIM:SG2   | KIM:SG6   | RTSE:SG2   | RTSE:SG3  | RTSE:SG3  | VAR:SG2 - SG4   | RISK:SG5   | EXD:SG3   | EXD:SG3  | EXD:SG4   | EXD:SG4  |
| <b>RMM</b>        | MON:SG2  | COMP:SG3 - SG4  | OTA:SG4  | KIM:SG6   | -   | -   | -   | RTSE:SG3   | -   | -   | MON:SG2   | -  | -   | -  | -   | RRM:SG1  |



| Assurance PRM  | SAFEcode   | MS SDL  | Open SAMM  | BSIMM   |
|--|--|---|--|---|
| <ul style="list-style-type: none"> <li>•Establish and maintain the strategic assurance training needs of the organization</li> <li>•Ensure resources have the training needed to do their job</li> </ul> | <ol style="list-style-type: none"> <li>1. Foundational (everyone)</li> <li>2. Advanced (secure coding and testing practices)</li> <li>3. Specialized (role-based)</li> </ol> | <ol style="list-style-type: none"> <li>1. Basic Concepts</li> <li>2. Common Baseline</li> <li>3. Custom Training</li> </ol> | <ol style="list-style-type: none"> <li>1. Technical Security Awareness training</li> <li>2. Role specific guidance</li> <li>3. Comprehensive security training and certifications</li> </ol> | <ol style="list-style-type: none"> <li>1. Create the software security satellite</li> <li>2. Make customized, role-based training available on demand</li> <li>3. Provide recognition for skills and career path progression</li> </ol> |



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



# SOFTWARE ASSURANCE FORUM

## BUILDING SECURITY IN

### SwA Self-Assessment (High Level)

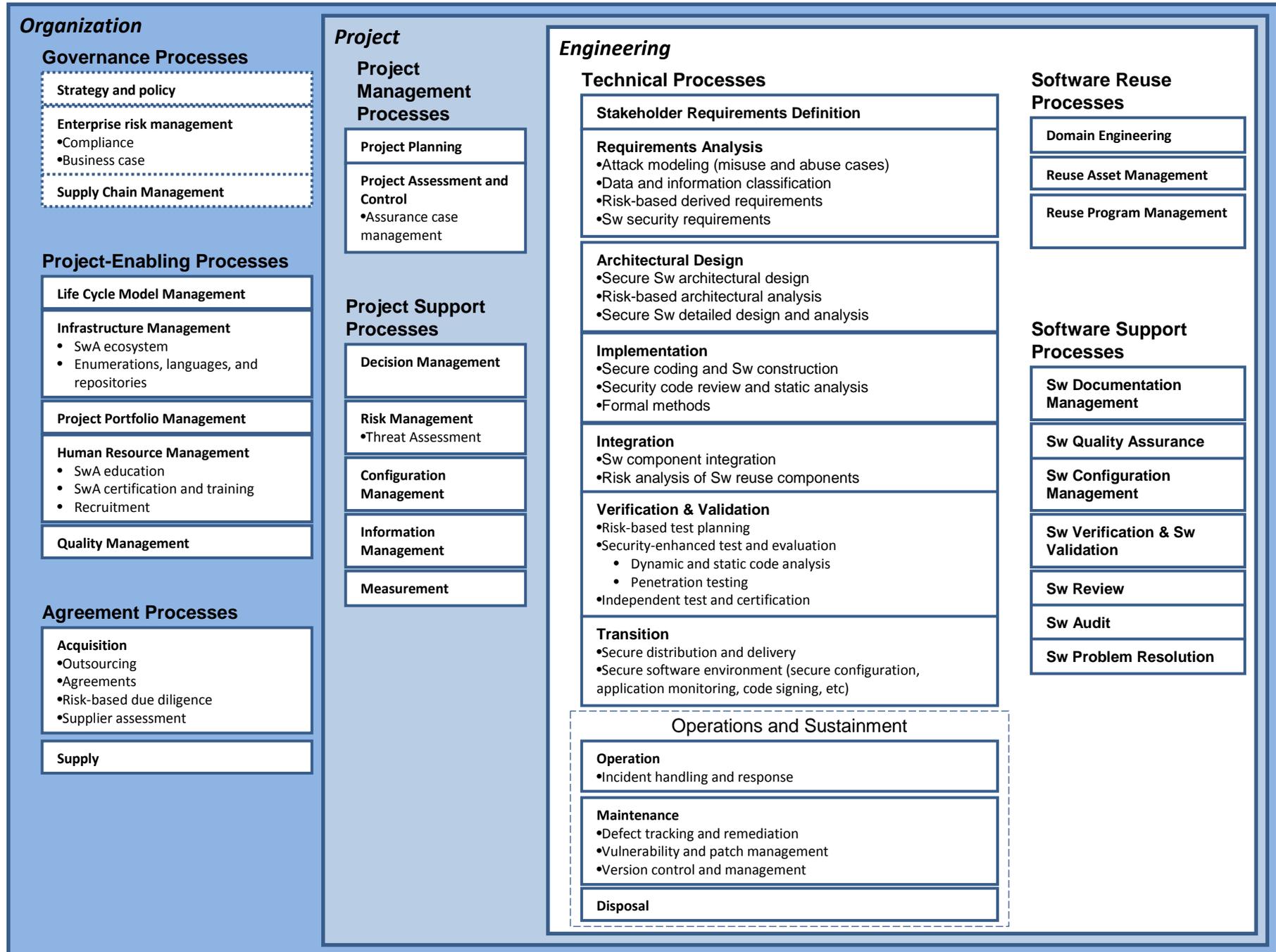
| Role  | Goal  | Expected Practice  | Activities   | Source       | BSIMM        | CMMI-ACQ | OSAM           | RMM          | MS SDL       | Developer Considerations | Acquirer Considerations | Practice Implementation Level | Notes |  |  |
|---|---|--|--|--------------|--------------|----------|----------------|--------------|--------------|--------------------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

Page 1



- Post the Updated Assurance Process Reference Model (PRM) Goals and Practices for comment
- Validate Mappings with authors of the common practices
- Expand the Assurance PRM to include operations
  - Collaborate with MAEC efforts
- Expand the mappings to include additional references and ensure alignment with emerging efforts
  - NIST Pubs (i.e. IR 7622, Risk Management, Developmental Security, Security Controls)
  - Cyber Scope
  - SAFECODE
  - Work items and standards from ISO (others?)
  - Other efforts that would inform the SwA Self-Assessment
- Continue discussions at future SwA events
- Understanding the synergies with the SwA Self Assessment and efforts to inform Acquisition Decisions

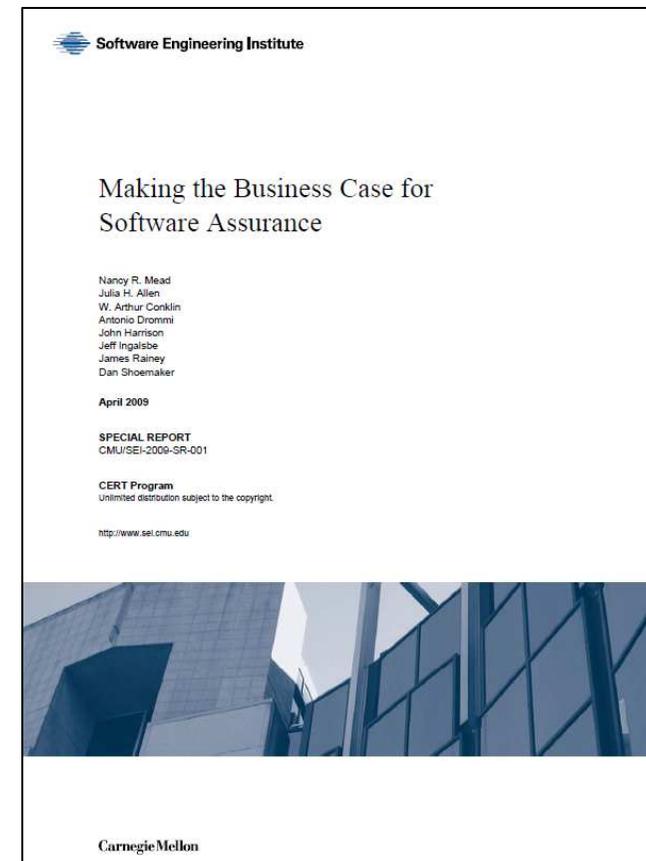
# Life-Cycle Standards View Categories (ISO/IEC 15288 and 12207)





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





Oct 08 → Feb 09 → May 09 →

**Practical Measurement Framework for Software Assurance and Information Security**

**Oct 2008**



The Center for Internet Security

The CIS Security Metrics

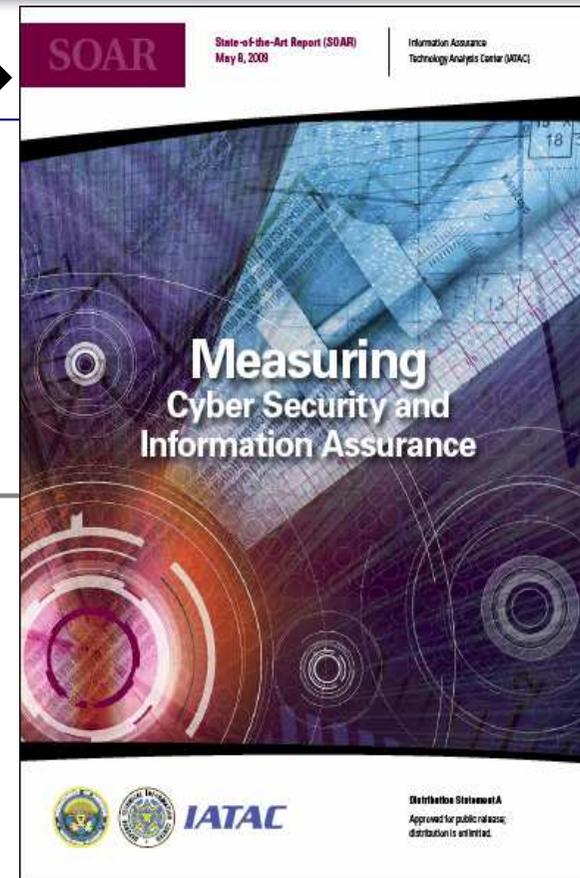
February 9

**2009**

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

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

Consensus Metric Definitions



# Measurement Guidance: Purpose

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

# Measurement Guidance: Scope & Resources

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





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)

## **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).



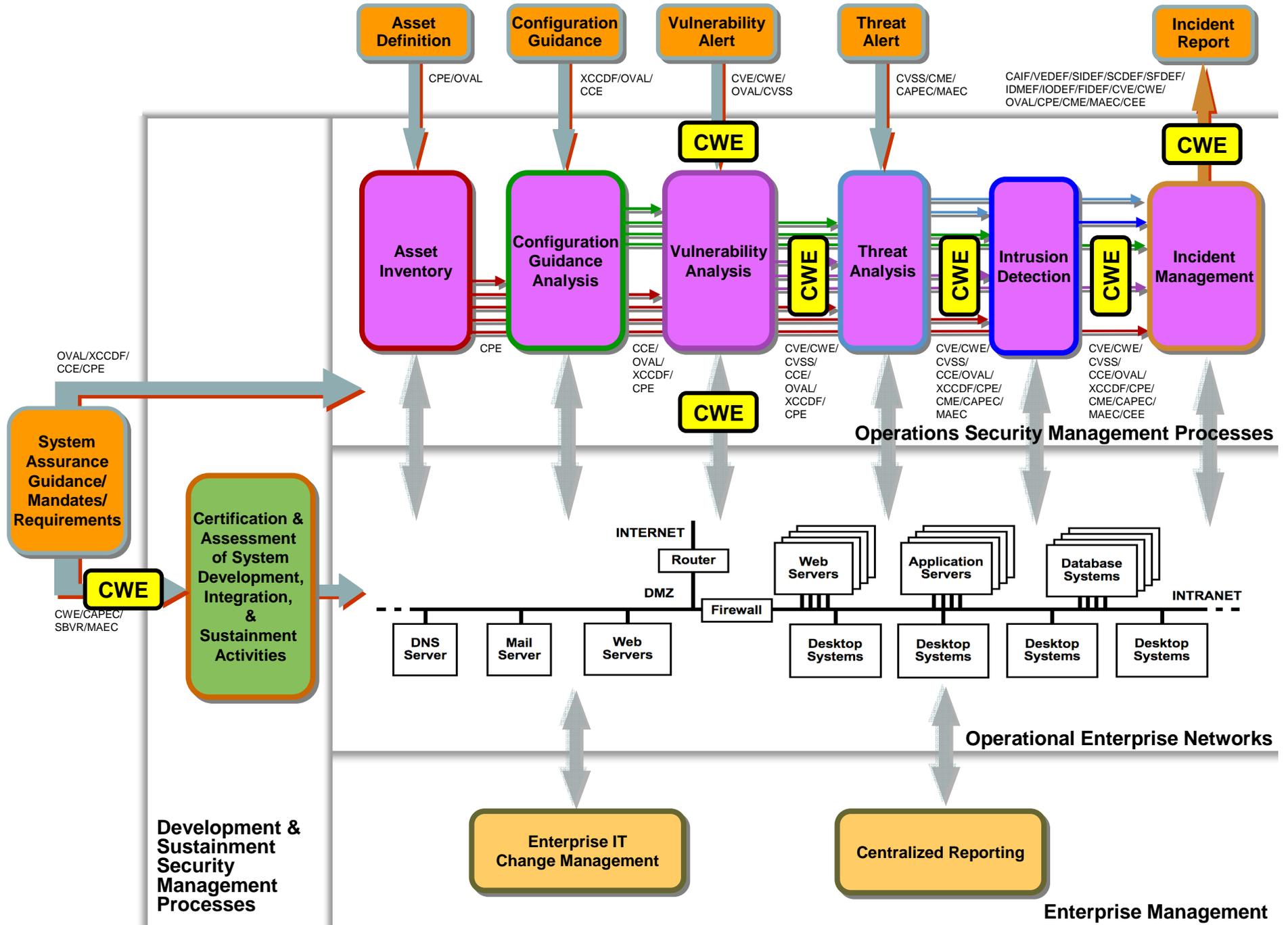
### **CWE-209: Error Message Information Leak**

- » 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



# 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**

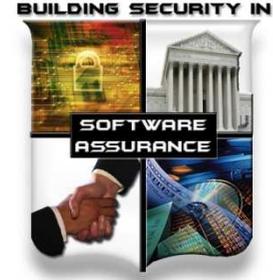


# Software Assurance Ecosystem: Turning Challenges into Solutions

- ▶ 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

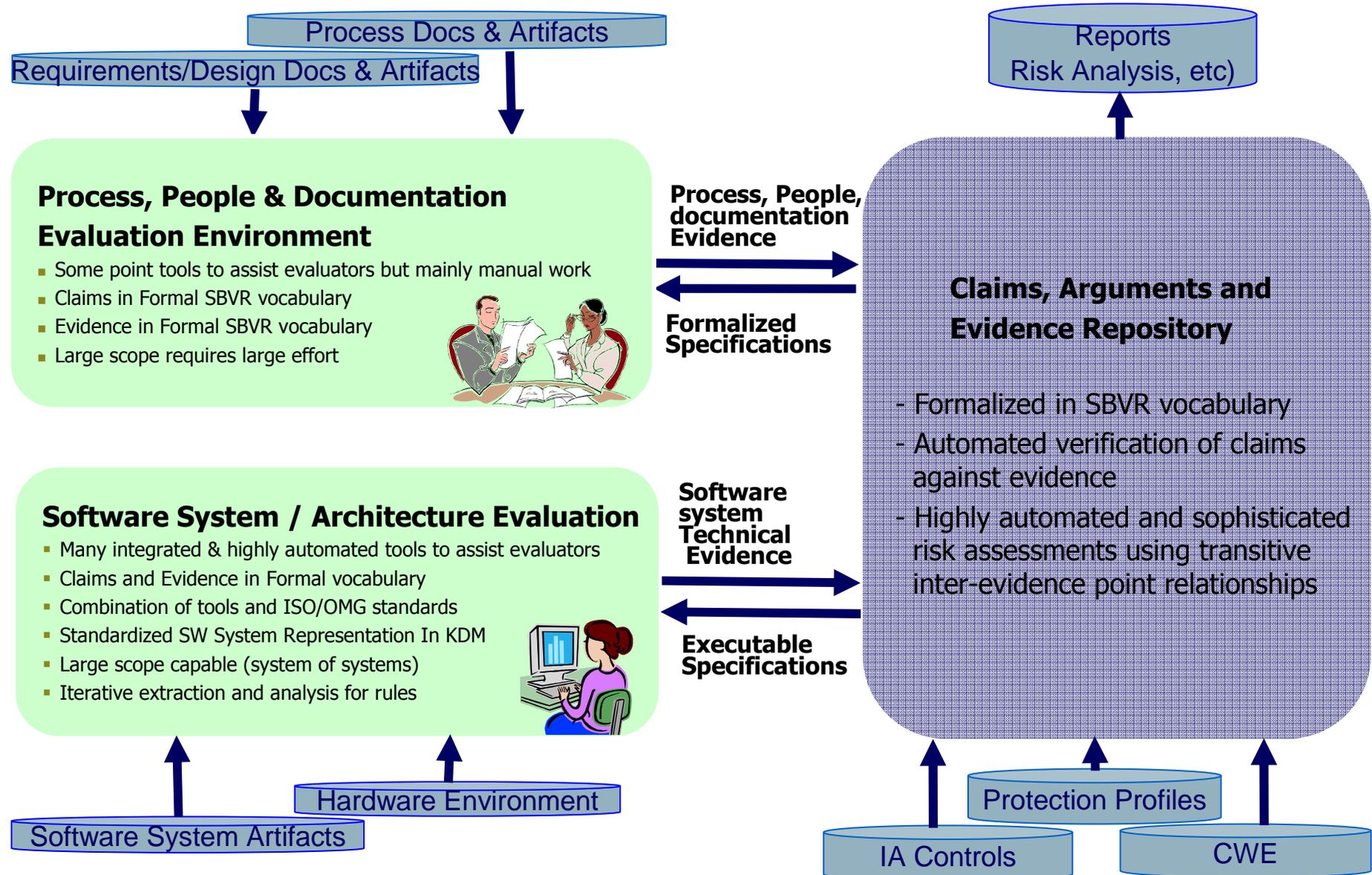


- ▶ Software Assurance Ecosystem enables industry and government to **leverage** and **connect** existing standards, policies, practices, processes and tools, in an affordable and efficient manner
- ▶ The key enabler is the Software Assurance (SwA) Ecosystem Infrastructure
  - an open standard-based 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



# Software Assurance Ecosystem: The Formal Framework

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





**ISO/IEC JTC 1/SC 27 NXXXX**  
**ISO/IEC JTC 1/SC 27/WG x NXXXXX**  
 REPLACES: N

**ISO/IEC JTC 1/SC 27**  
**Information technology - Security techniques**  
 Secretariat: DIN, Germany

**DOC TYPE:** NB MWI Proposal for a technical report (TR)  
**TITLE:** National Body New Work Item Proposal on "Secure software development and evaluation under ISO/IEC 15406 and ISO/IEC 18405"  
**SOURCE:** INCITSACS 1, National Body of (US)  
**DATE:** 2006-09-30  
**PROJECT:** 15406 and 18405  
**STATUS:** This document is circulated for consideration at the forthcoming meeting of SC 27/WG 3 to be held in Redmond (WA, USA) on 2<sup>nd</sup> - 6<sup>th</sup> November 2006.  
**ACTION ID:** ACT  
**DUE DATE:**  
**DISTRIBUTION:** P, D and L-Members  
 W: Larry Sid 27 Chairman  
 M: De Soete, SC 27 Vice-Chair  
 E: J. Humphreys, K. Naraenara, Y. Bultin, M.-C. Kang, K. Rammberg, WG-Consensus  
**MEDIUM:** Live Intra-server  
**NO. OF PAGES:** xx

Secretariat ISO/IEC JTC 1/SC 27 -  
 DIN Document Institut für Normung e. V., Burgstrasse 6, 10772 Berlin, Germany  
 Telephone: +49 30 201-15952; Facsimile: +49 30 2501-7233; E-mail: [iso@iso.din.de](mailto:iso@iso.din.de)  
[HTTP://www.iso/iso/secretariat/](http://www.iso/iso/secretariat/)

**Common Criteria v4 CCDB**  
**•TOE to leverage CAPEC & CWE**  
**•Also investigating how to leverage ISO/IEC 15026**  
**NIAP Evaluation Scheme**  
**•Above plus**  
**•Also investigating how to leverage SCAP**

**New Work Item Proposal**  
**NP submitting**  
**PROPOSAL FOR A NEW WORK ITEM**

|  |   |
|--|---|
| Date of presentation of proposal<br>YYYY-MM-DD | Proposer: ISO/IEC JTC 1/SC 27                 |
| Secretariat<br>National Body                   | ISO/IEC JTC 1 N XXXX<br>ISO/IEC JTC 1/SC 27 N |

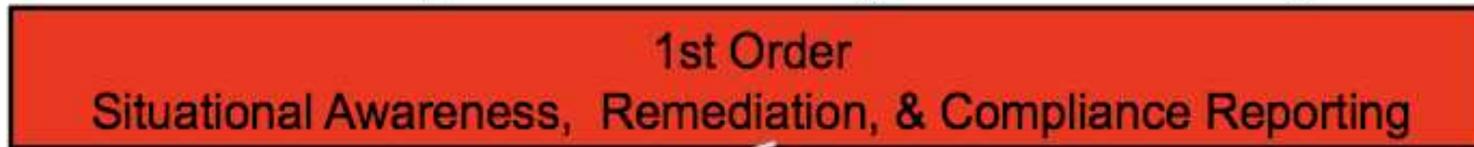
**A proposal for a new work item shall be submitted to the secretariat of the ISO/IEC joint technical committee concerned with a copy to the ISO Central Secretariat.**  
**Presentation of the proposal**  
**Title** Secure software development and evaluation under ISO/IEC 15406 and ISO/IEC 18405

**Scope**  
 In the case where a target of evaluation (TOE) being evaluated under ISO/IEC 15406 and ISO/IEC 18405, includes specific software portions, the TOE developer may optionally present the developer's technical rationale for mitigating software common attack patterns and related weaknesses as described in the latest revision of the Common Attack Pattern Enumeration and Classification (CAPEC) model table from <http://capec.mitre.org/>. The developer's technical rationale is expected to include a range of mitigation techniques, from architectural properties to design features, coding techniques, use of tools or other means.

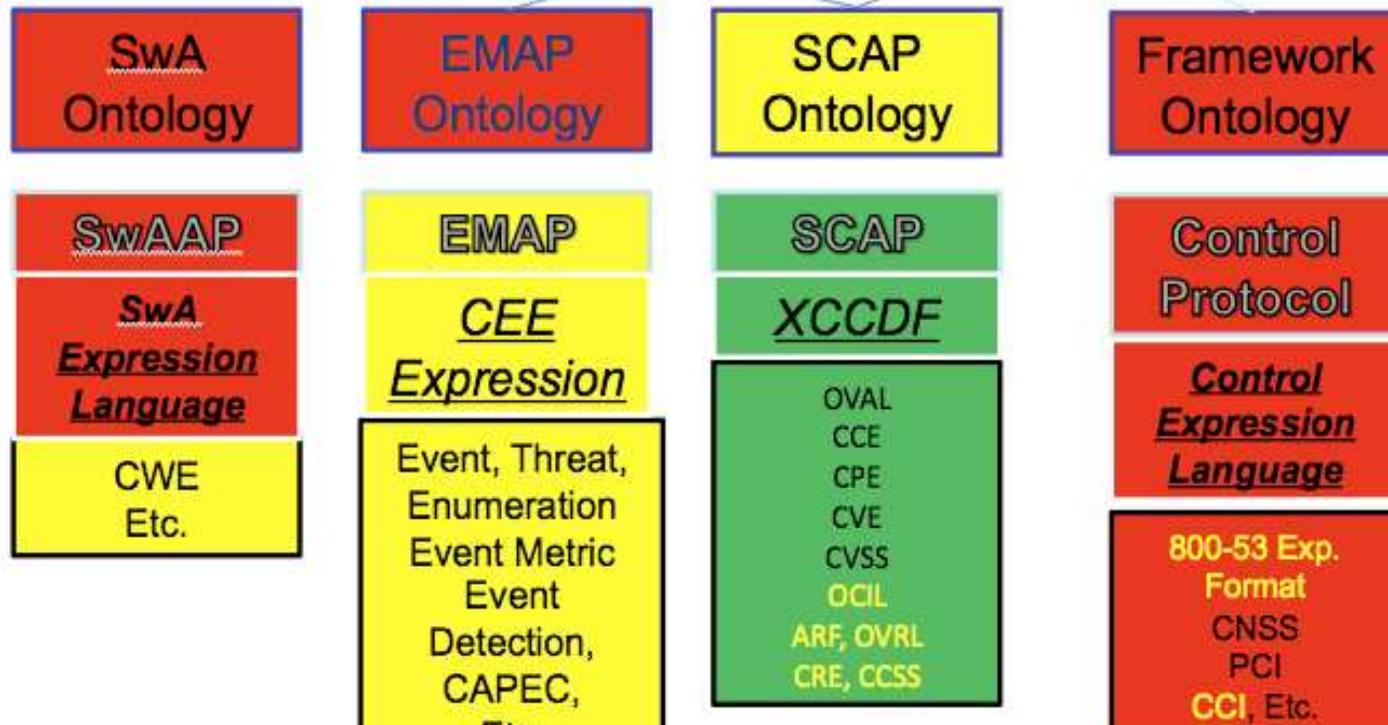
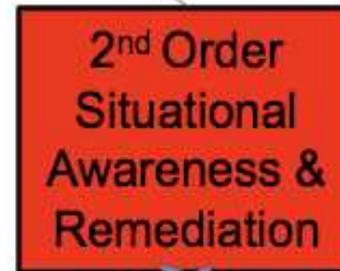
**The Technical Report (TR) provides guidance for the developer and the evaluator on how to use the CAPEC as a technical reference point during the TOE development phase and in an evaluation of the TOE secure software under ISO/IEC 15406 and 18405, by addressing:**

- A refinement of the IS 15406 Attack Potential evaluation table for software, taking into account the entries contained in the CAPEC and their characteristic.
- How the information for mitigating software common attack patterns and related weaknesses is used in an IS 15406 evaluation, in particular providing guidance on how to determine which attack patterns and weaknesses are applicable to the TOE, taking into consideration of
  - the TOE technology;
  - the TOE security problem definition;
  - the interfaces the TOE exports that can be used by potential attackers;
  - the Attack Potentials that the TOE needs to provide resistance for.
- How the CAPEC and related Common Weakness Enumeration (CWE) weaknesses are used by the evaluator, who needs to consider at the applicant's attack patterns and be able to exploit specific related software weaknesses while performing the subsequent vulnerability analysis (AVA\_VAN) activities on the TOE.
- How incomplete entries from the CAPEC are resolved during an IS 15406 evaluation.
- How the evaluator's attack and weakness analysis of the TOE incorporates other attacks and weaknesses not yet documented in the CAPEC.

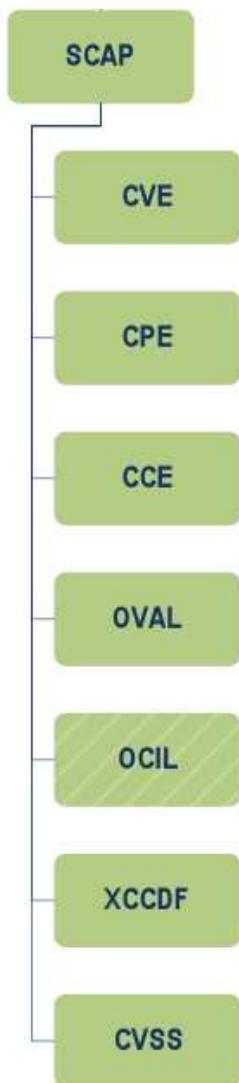
**The TR also investigates specific elements from the ISO/IEC 15026 (and its revision) are applicable to the guidelines being developed in the TR within the context of IS 15406 and 18405.**



*Receive 1<sup>st</sup> order for 'free' by virtue of 2<sup>nd</sup> order ontologies mapped through SCAP.*



NIST Validation



SCAP 1.1 uses the following specifications:

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

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

Special Publication 800-126  
Revision 1 (DRAFT)

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

Recommendations of the National Institute  
of Standards and Technology

Stephen Quinn  
David Waltermire  
Christopher Johnson  
Karen Scarfone  
John Banghart

|           |  |            |
|-----------|--|------------|
| <b>4.</b> | <b>SCAP General Requirements and Conventions</b> .....       | <b>4-1</b> |
| 4.1       | Support for Legacy SCAP Versions.....                        | 4-1        |
| 4.2       | XCCDF Conventions and Requirements .....                     | 4-1        |
| 4.2.1     | Metadata Elements .....                                      | 4-1        |
| 4.2.2     | Use of CPE Names .....                                       | 4-2        |
| 4.2.3     | The <xccdf:Benchmark> Element .....                          | 4-3        |
| 4.2.4     | The <xccdf:Profile> Element.....                             | 4-3        |
| 4.2.5     | The <xccdf:Rule> Element.....                                | 4-4        |
| 4.2.6     | Allowed Check System Usage .....                             | 4-5        |
| 4.2.7     | XCCDF Test Results.....                                      | 4-10       |
| 4.3       | OVAL Conventions and Requirements.....                       | 4-12       |
| 4.3.1     | Supported Previous Versions of OVAL (5.3, 5.4, and 5.5)..... | 4-13       |
| 4.3.2     | Support for Deprecated Constructs in OVAL .....              | 4-13       |
| 4.3.3     | OVAL Schema Specification .....                              | 4-13       |
| 4.3.4     | OVAL Results .....   | 4-13       |
| 4.4       | OCIL Conventions .....                                       |            |
| 4.5       | CPE Conventions .....  |            |
| 4.6       | CCE Conventions.....   |            |
| 4.7       | CVE Conventions .....  |            |
| 4.8       | CVSS Conventions.....  |            |

---

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

---

Recommendations of the National Institute  
of Standards and Technology

---

Stephen Quinn  
David Waltermire  
Christopher Johnson  
Karen Scarfone  
John Banghart

---

|           |   |            |
|-----------|---|------------|
| <b>5.</b> | <b>SCAP Use Case Requirements.....</b>                    | <b>5-1</b> |
| 5.1       | SCAP Data Streams.....                                    | 5-1        |
| 5.2       | SCAP Configuration Verification.....                      | 5-1        |
| 5.3       | SCAP Vulnerability Assessment.....                        | 5-3        |
| 5.3.1     | SCAP Vulnerability Assessment Using XCCDF and OVAL .....  | 5-3        |
| 5.3.2     | SCAP Vulnerability Assessment Using Standalone OVAL ..... | 5-4        |
| 5.3.3     | OVAL Definitions and Vulnerability Assessment.....        | 5-4        |
| 5.4       | Patch Validation .....                                    | 5-4        |
| 5.4.1     | Using OVAL Definitions for Patch Validation .....         | 5-5        |
| 5.4.2     | Referencing an OVAL Patch Data Stream.....                |            |
| 5.5       | SCAP Inventory Collection .....                           |            |

---

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

---

Recommendations of the National Institute  
of Standards and Technology

---

Stephen Quinn  
David Waltermire  
Christopher Johnson  
Karen Scarfone  
John Banghart

---



# Software Assurance Automation Protocol (**SwAAP**)

- For measuring & enumerating software weaknesses and the assurance cases.

Common Weakness Enumeration (**CWE**),

Common Attack Pattern Enumeration & Classification (**CAPEC**),

Malware Attribute Enumeration & Characterization (**MAEC**),

Common Weakness Scoring System (**CWSS**),

Software Assurance Findings Expression Schema (**SAFES**),

NIST SAMATE's "Software Transparency Label",

ISO/IEC 15026 "Assurance Case" (**ISO 15026**),

OMG Software Assurance Evidence Metamodel (**OMG SAEM**),

OMG Argumentation Metamodel (**OMG ARG**),

OMG Structured Metrics Metamodel (**OMG SMM**),

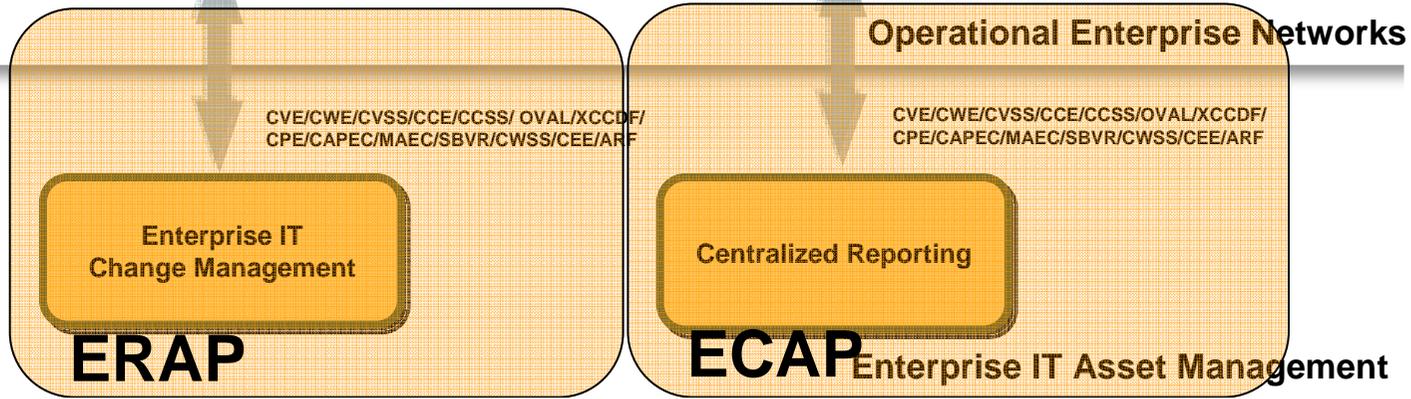
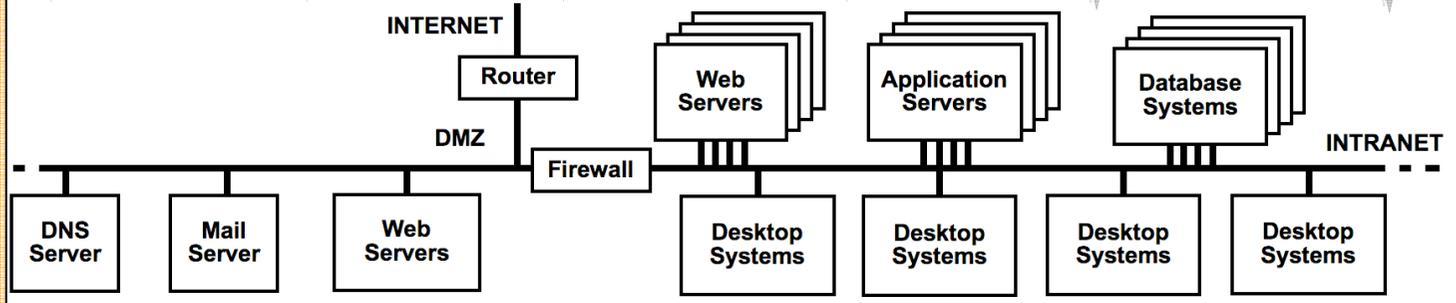
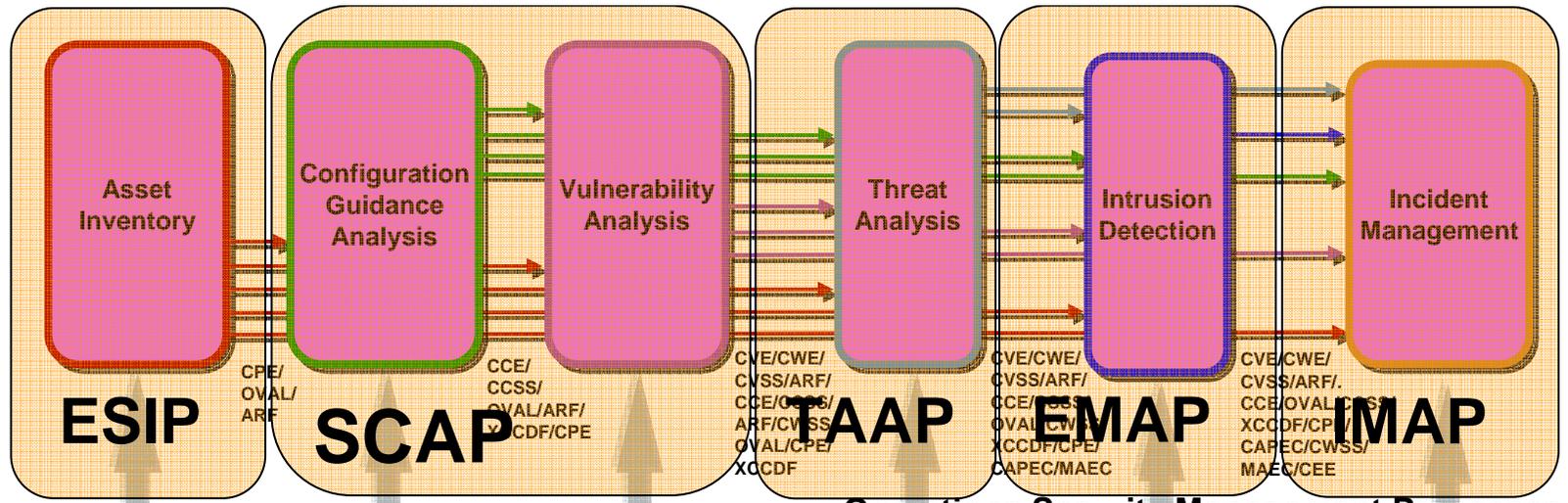
OMG Knowledge Discovery Metamodel (**OMG KDM**),

OMG Abstract Syntax Tree Metamodel (**OMG ASTM**)

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

# “Other” Automation Protocols (“O”AP)

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



**Development & Sustainment Security Management Processes**

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



# 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

# SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN



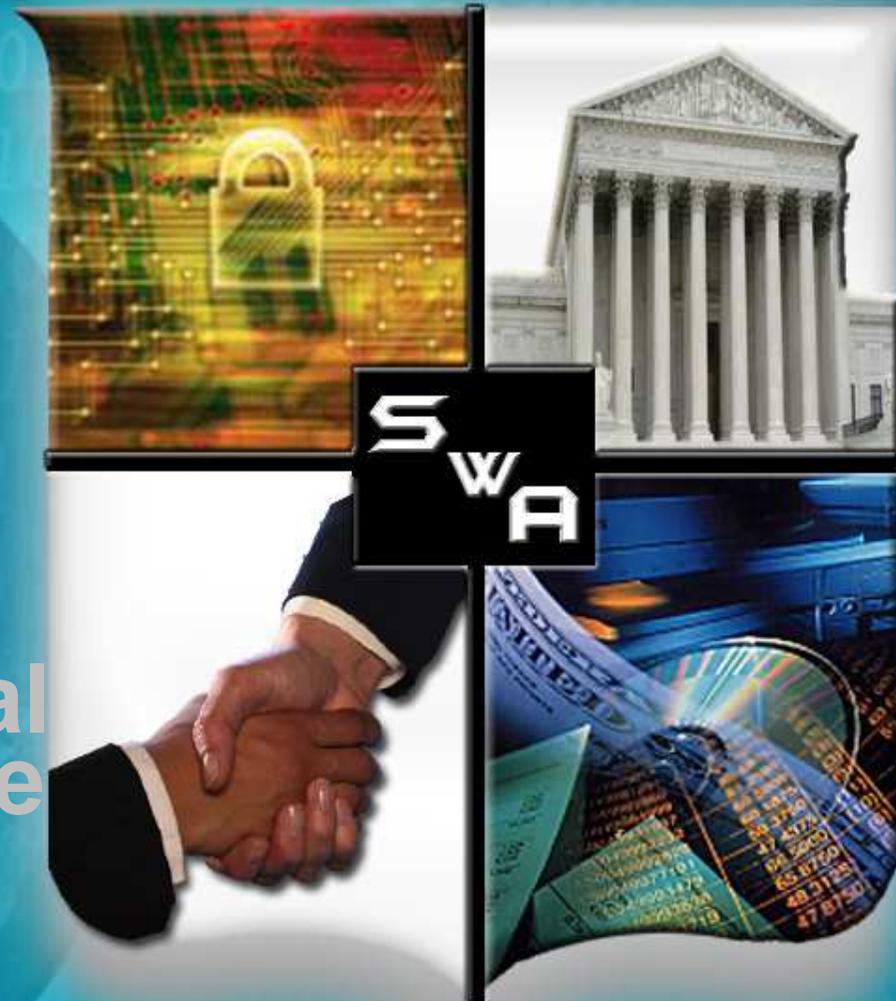
Homeland  
Security



Commere



National  
Defense



SWA

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