Understanding SwA Supply and Demand (Development)

SwA Working Groups June 22, 2010

Michele Moss, Booz Allen Hamilton

Ed Wotring, Information Security Solutions

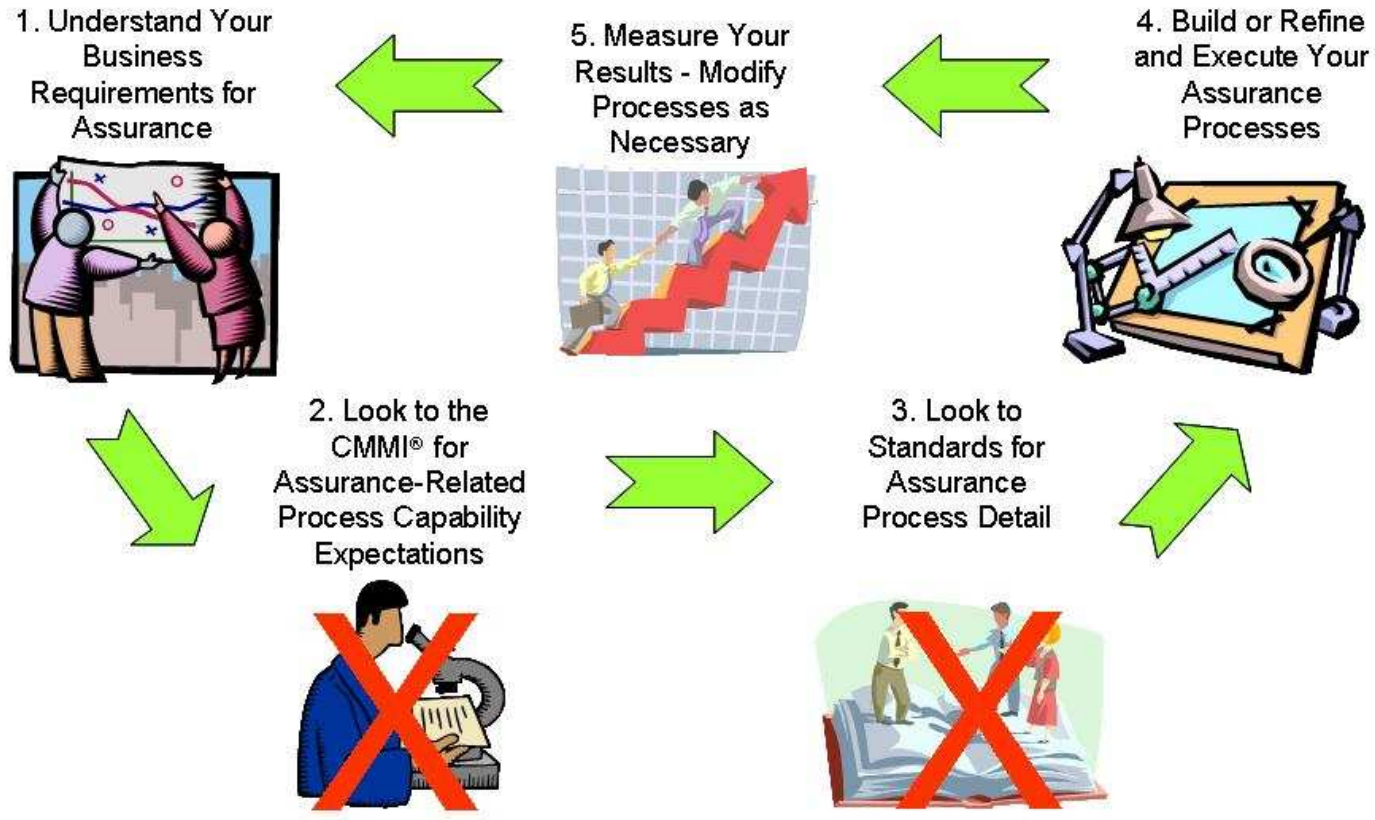
- Overview Of Challenges In The Implementation Of SwA Practices
- Understanding Practice Implementation (A Self Assessment Approach)
- Leveraging The Practice Implementation Self Assessment During Acquisition

- 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

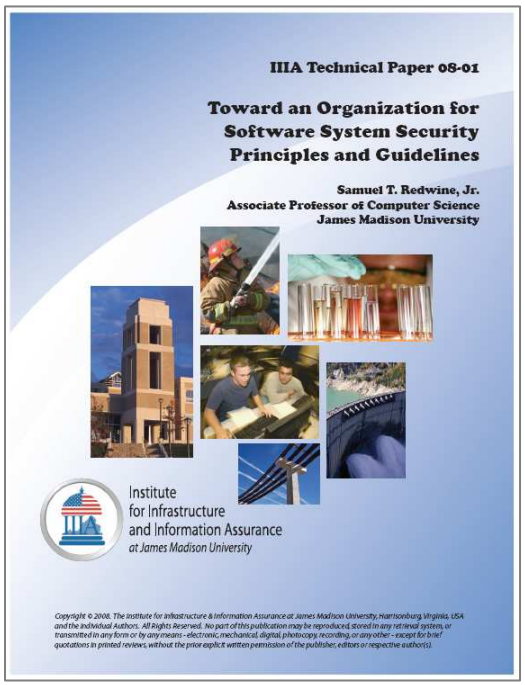
Homeland Security

- 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

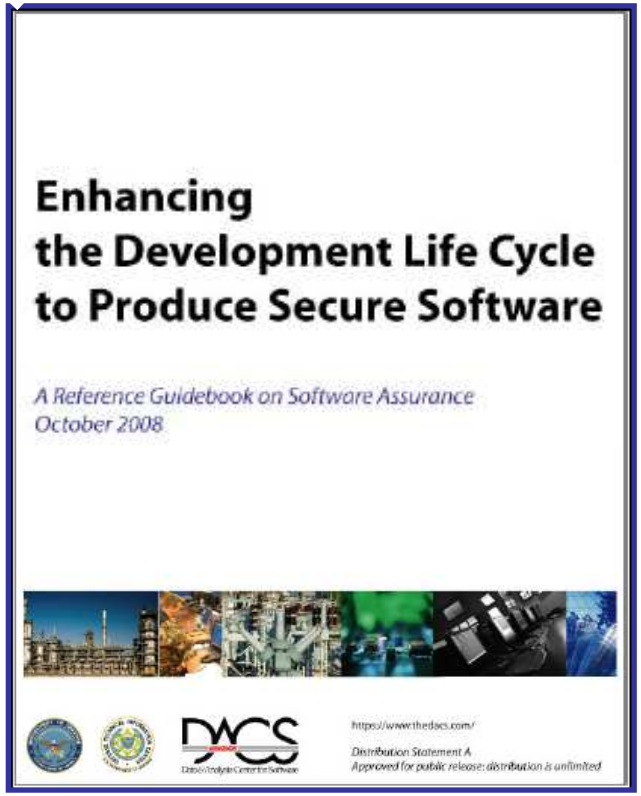


Courtesy of Paul Croll



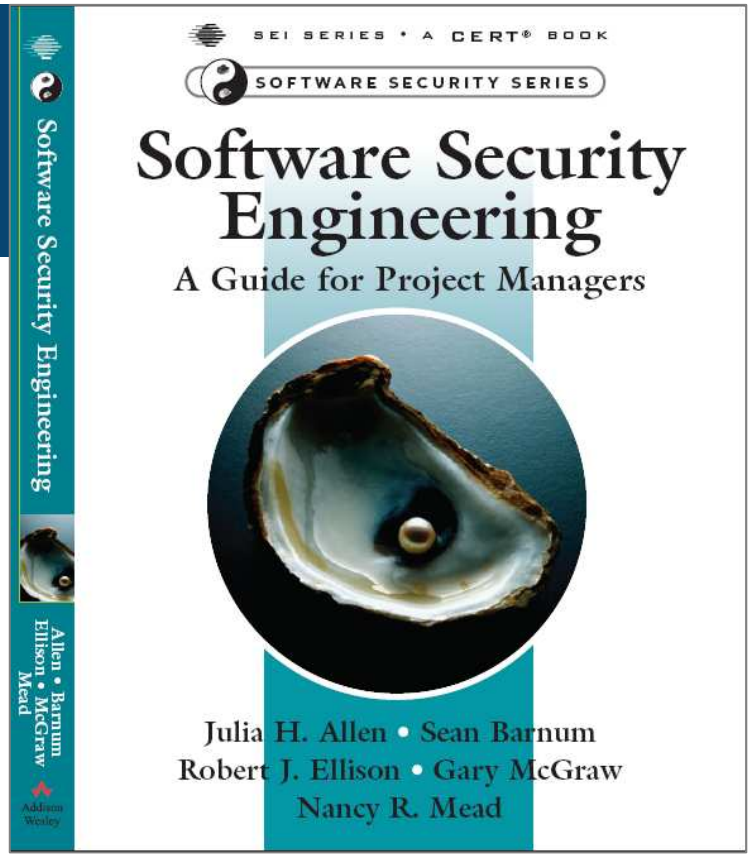
https://buildsecurityin.us-cert.gov/swa/wetwgdocs.html

- Does provide information to help readers understand, assess, and choose from among the growing number of security-enhancing SDLC processes, methodologies, practices, techniques, and supporting tools
- Does not espouse a specific approach or philosophy.
- Does not attempt to evaluate or critique security-enhancement approaches

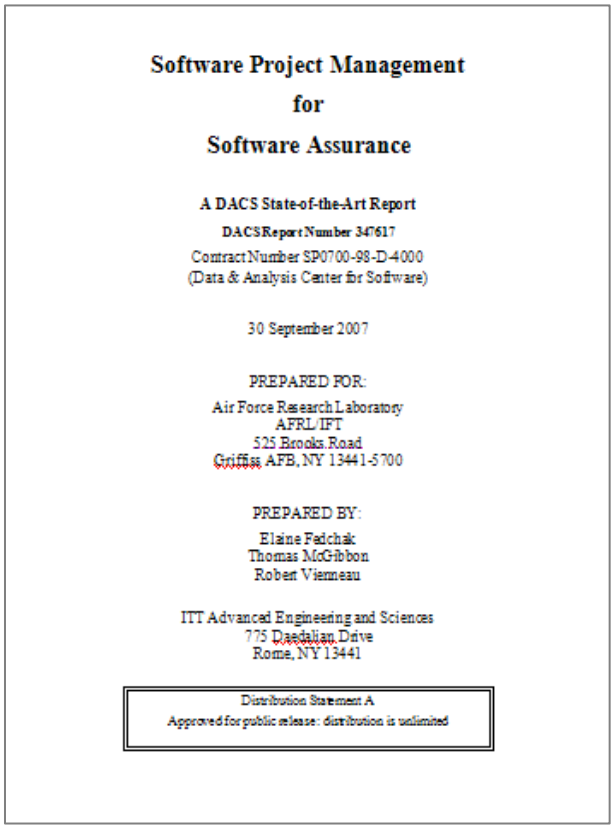


https://www.thedacs.com/techs/enhanced_life_cycles/

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



- The primary audience for this report is software project managers
- Information on how the need for software assurance affects software project management
- Tools and resources for quantifying the effects of software assurance on software development, both in terms of planning (cost estimation and budgeting), and in terms of overall cost-effectiveness and return on investment
- DACS Report Number 347617



Software Project Management
for
Software Assurance

A DACS State-of-the-Art Report
DACS Report Number 347617
Contract Number SP0700-98-D-4000
(Data & Analysis Center for Software)

30 September 2007

PREPARED FOR:
Air Force Research Laboratory
AFRL/IFT
525 Brooks Road
Griffiss AFB, NY 13441-5700

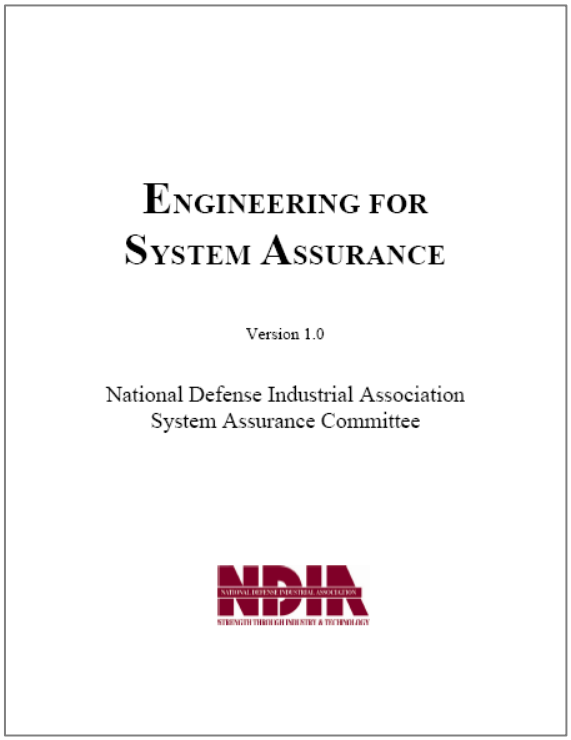
PREPARED BY:
Elaine Fedchak
Thomas McGibbon
Robert Vienneau

ITT Advanced Engineering and Sciences
775 Daedalian Drive
Rome, NY 13441

Distribution Statement A
Approved for public release: distribution is unlimited

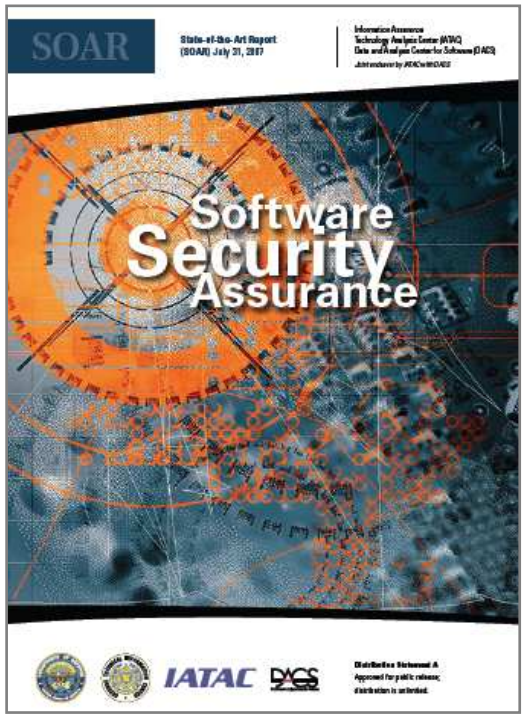
https://acc.dau.mil/CommunityBrowser.aspx?id=219497

- NDIA/DoD guidebook providing process and technology guidance to increase the level of system assurance.
- Intended primarily to aid program managers (PMs) and systems engineers (SEs) who are seeking guidance on how to incorporate assurance measures into their system life cycles.



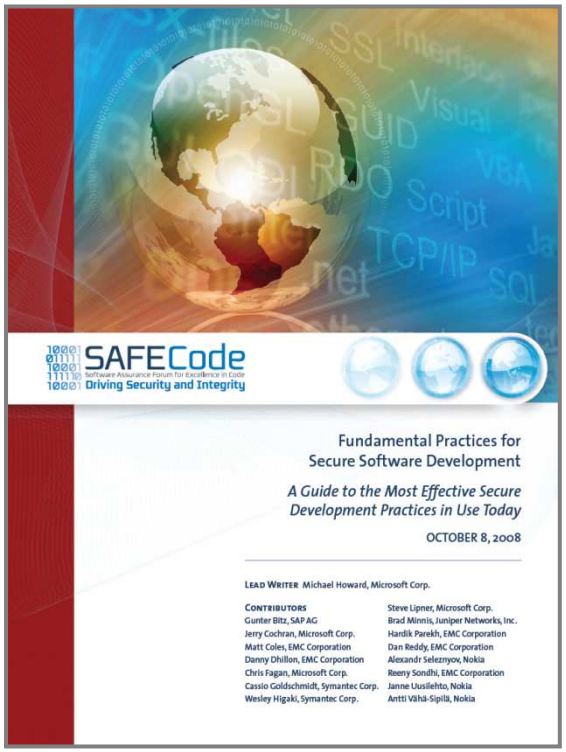
http://www.acq.osd.mil/sse/ssa/docs/SA-Guidebook-v1-Oct2008.pdf

- Describes numerous methodologies, best practices, technologies, and tools currently being used to specify, design, and implement software that will be less vulnerable to attack, and to verify its attack-resistance, attack-tolerance, and attack-resilience;
- Offers a large number of available print and online resources from which readers can learn more about the principles and practices that constitute Software Security Assurance;
- Provides observations about potentials for success, remaining shortcomings, and emerging trends across the S/W Security Assurance landscape.



http://iac.dtic.mil/iatac/download/security.pdf

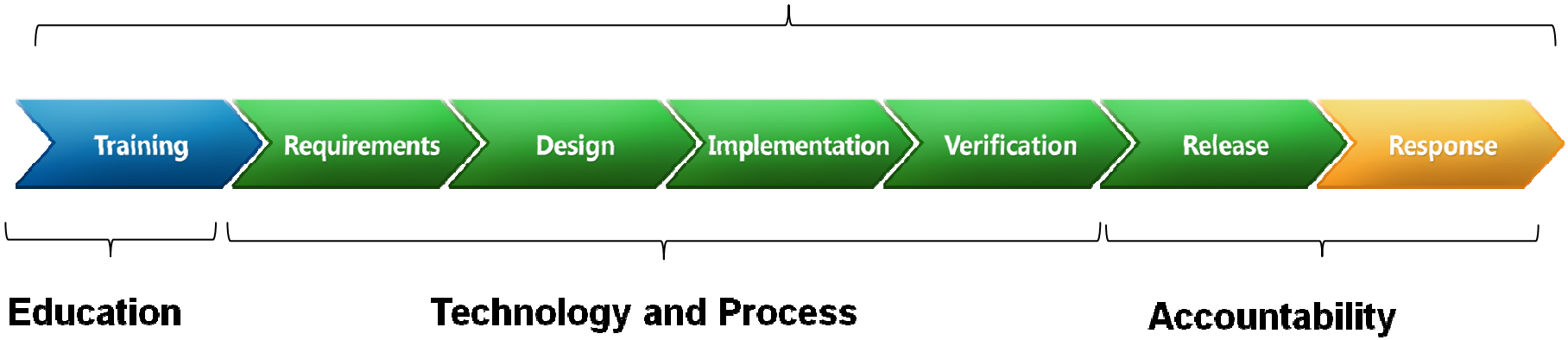
- Fundamental Practices for Secure Software Development: Guide to the Most Effective Secure Development Practices in Use Today, Oct 8, 2008
  - Common security-related elements of software development methodologies
  - Secure Programming practices:
  - Test to validate robustness and security
  - Code Integrity and Handling
  - Documentation (about software security posture & secure configurations)



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

Delivering secure software requires:

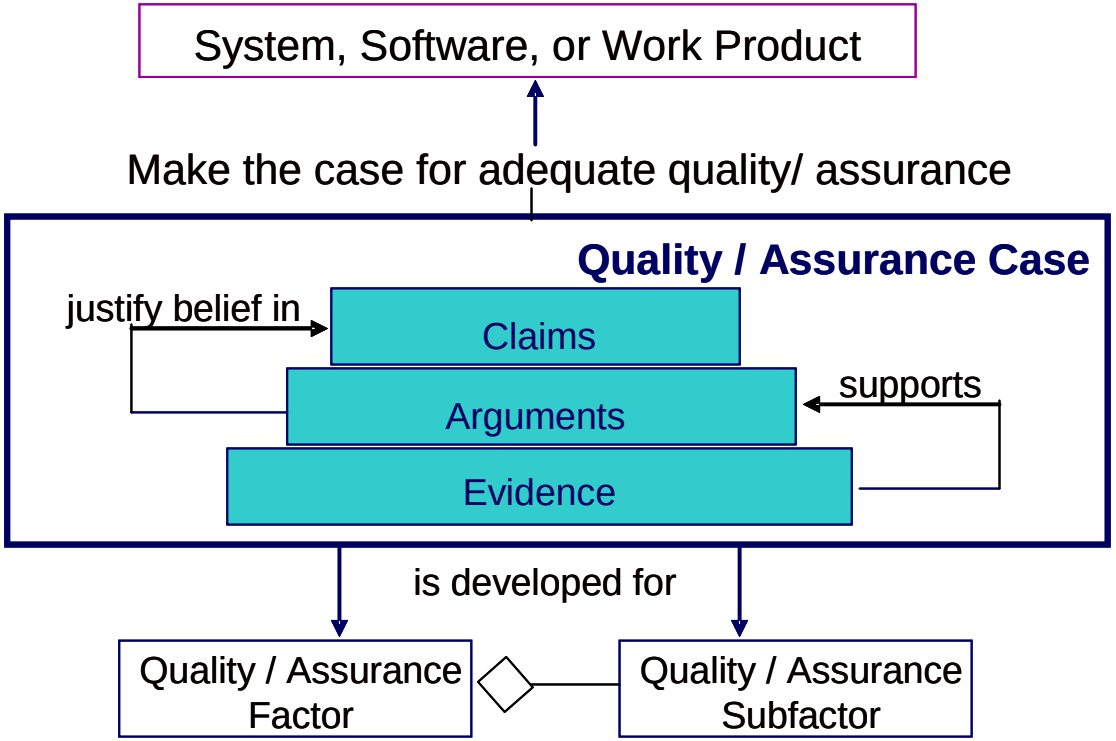
Executive commitment → SDL a mandatory policy at Microsoft since 2004



Ongoing Process Improvements → 6 month cycle

http://www.microsoft.com/sdl

An Assurance Case



Attributes

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

Adapted from a slide by Joe Jarzombek who, in turn, credited IEEE CS alternative proposal for 15026 and CMU SEI QUASAR tutorial by Donald Firesmith, March 2007

Requirements

What is wanted

What is created

Unmet requirements

Extra Requirements

Quality - Does the result meet the requirements?

Assurance -

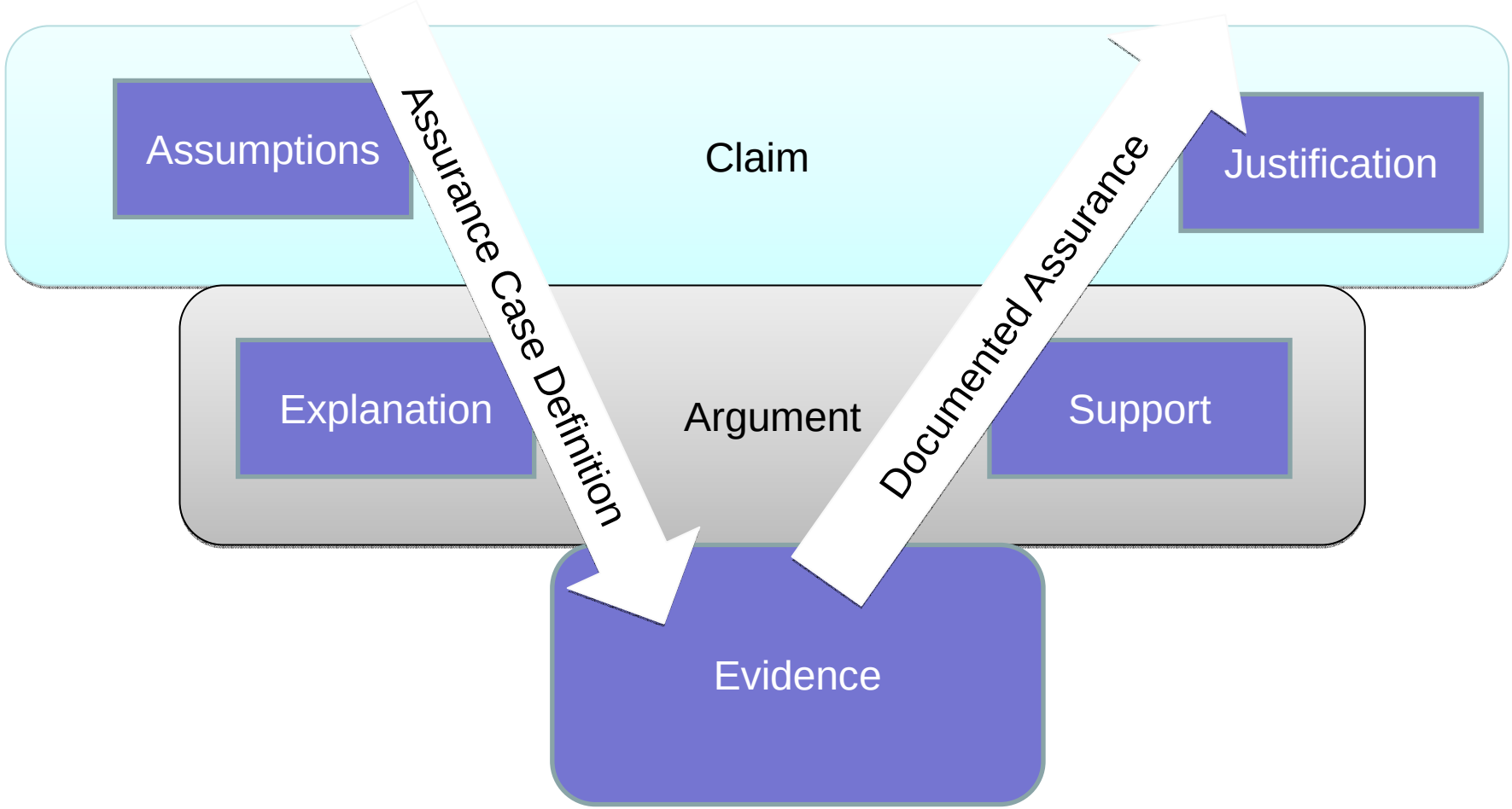
- What other features are enabled?
- How do these other features impact the original requirements?

It isn't about Quality OR Assurance …

It is about Quality AND Assurance

Courtesy of Margaret Nadworny and Michele Moss


SOFTWARE ASSURANCE FORUM
BUILDING SECURITY IN
Creating An Assurance Case


Assumptions
Claim
Justification
Assurance Case Definition
Documented Assurance
Explanation
Argument
Support
Evidence

- June 2007 – SwA P&P Working Group initiated efforts to collaborate with industry (SEI and ISSEA) to integrate security in capability based process improvement and capability benchmarking
- March 2007: SEPG Birds of a Feather
- August 7, 2007: Industry Assurance for CMMI ® Meeting
- September 2007: Motorola, Lockheed Martin and Booz Allen form Assurance Working Group
- October 2007: Assurance Harmonization Working Group
- January 2008: Assurance Focus Topic Working Group
- July 16, 2008: Gained CMMI ® Steering Group approval to create Focus Topic for Assurance
- February 27, 2009: Submitted Change Requests for consideration in CMMI v 1.3
- Updating Assurance PRM practices with refined practices, revised CMMI mapping, and industry LL

Courtesy of Margaret Nadworny and Michele Moss

- If there is a one size fits all solution, it must be at a level of detail that the context is applicable in diverse contexts (Defense, National Security, Finance, Heath care, Aviations, Telecommunications)
- Discomfort in using assurance for acquisition decisions
  - Potential source of liability – false sense of assurance
  - Integrity of appraisals – exaggerated claims
  - Potential misinterpretation of appraisal results - Cannot ensure that any product is secure
- Implementation of the current model is costly – cognizant of increased size/scope of model
- We don’t need another certification!
- Assurance must be built in

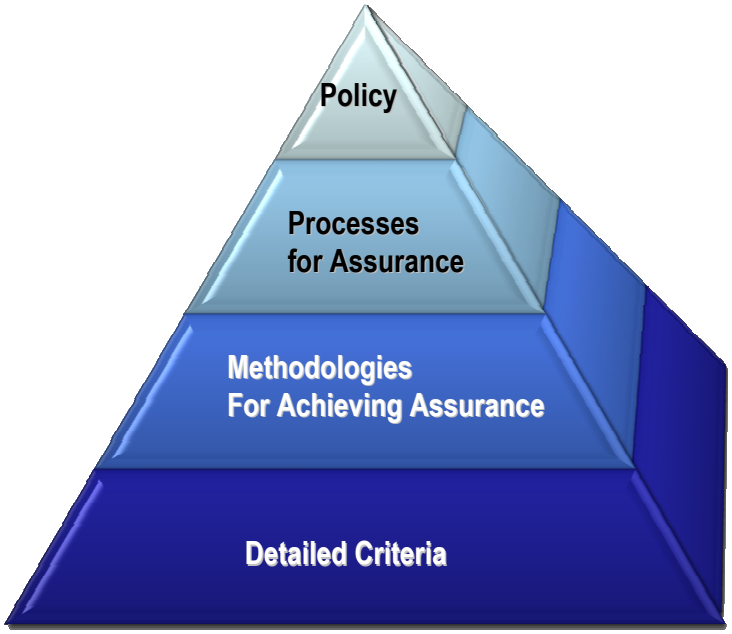
Courtesy of Margaret Nadworny and Michele Moss

- Objectives
  - Discuss "Best Practices" for Assurance
  - Identify sources of best practices for assurance
  - Understand Lessons Learned associated with use of assurance processes and practices
  - Understand stakeholder views for deploying practices and addressing assurance in CMMI®
- Participants
  - Government, Industry, Academia
  - Acquirers, vendors, developers, standards organizations, test labs, and research

Courtesy of Margaret Nadworny and Michele Moss

- Key references were in “draft” or a presentation/discussion
- The practices were not codified in a standard
- Solutions were being identified through “Research” and pilots
- The acquisition community was not requesting the practices – no demand
- Relied on assumptions that were not valid (raise awareness and they will act)
- Outreach efforts resulted – “So what do you want me to do?”
- Existing documentation was in SwA Community speak

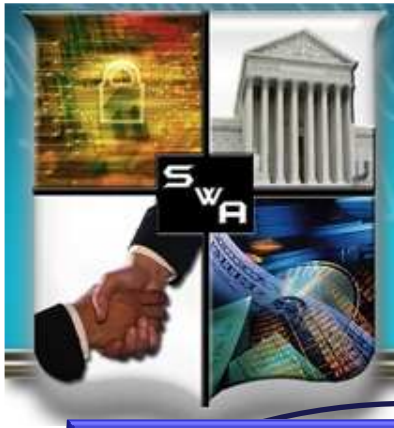
Project leadership and team members need to know where and how to contribute



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

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

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



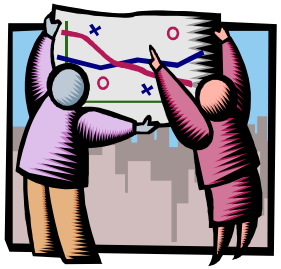
SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

Process Improvement Lifecycle - A Process for Achieving Assurance

Mission/Business Process

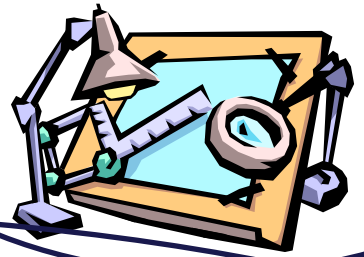
Understand Your Business Requirements for Assurance



Measure Your Results

Information System

Build or Refine and Execute Your Assurance Processes



Understand Assurance-Related Process Capability Expectations

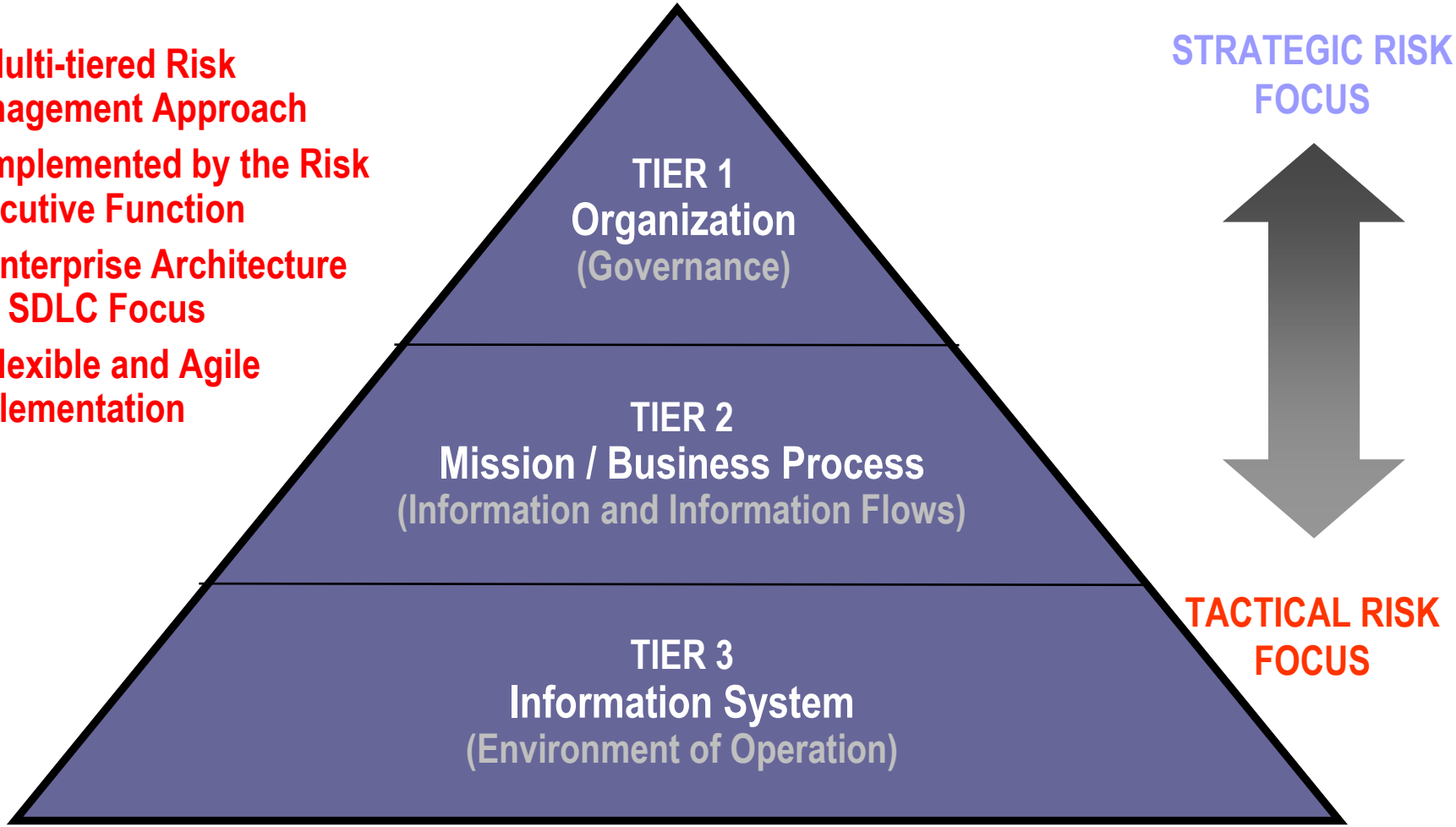
Organization Support

Look to Standards for Assurance Process Detail

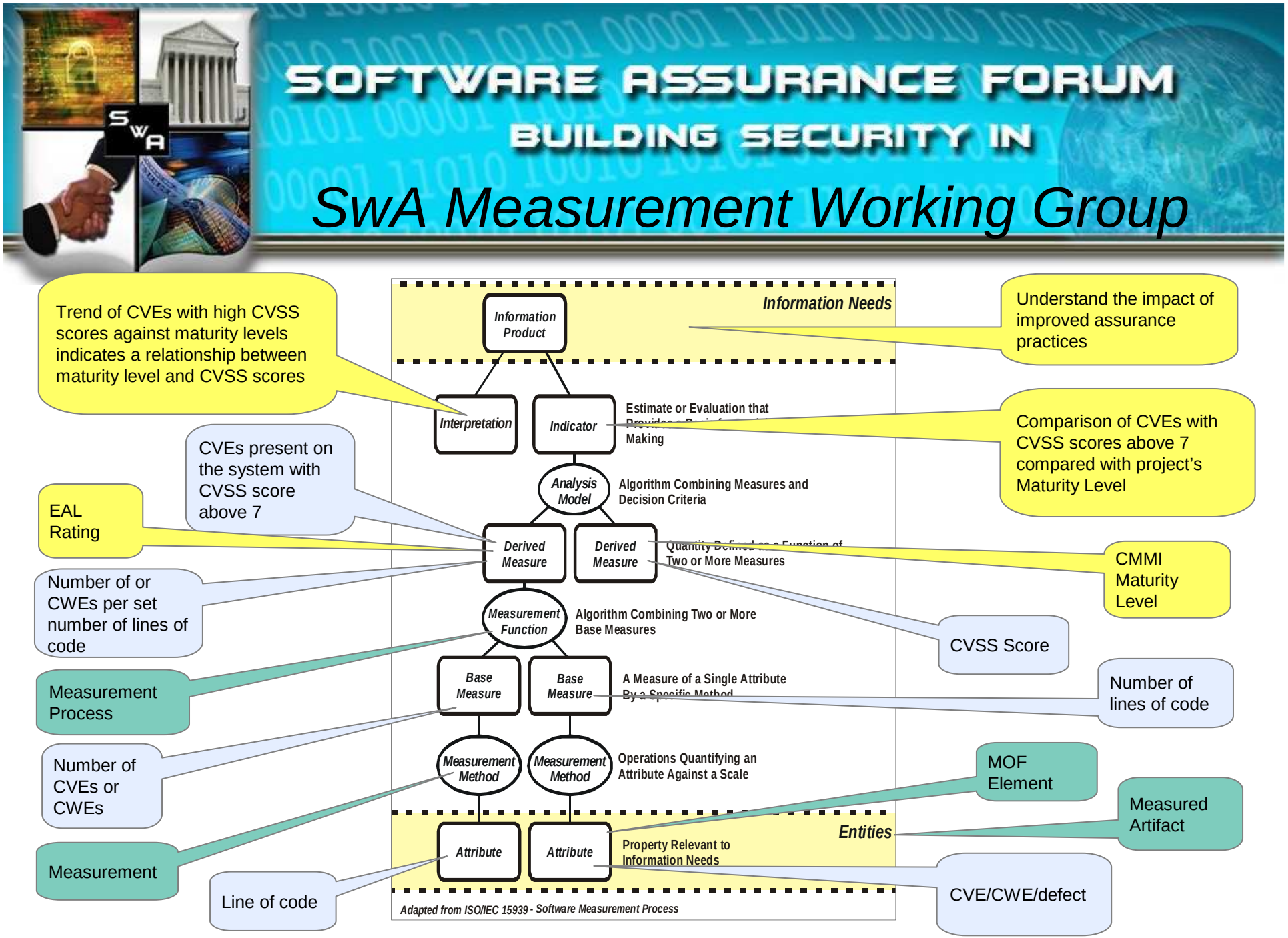
Adapted from: Paul Croll, Computer Sciences Corporation, August 2007

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


SOFTWARE ASSURANCE FORUM
BUILDING SECURITY IN
SwA Measurement Working Group
Trend of CVEs with high CVSS scores against maturity levels indicates a relationship between maturity level and CVSS scores
Information Needs
Information Product
Understand the impact of improved assurance practices
Interpretation
Indicator
Estimate or Evaluation that
Making
Comparison of CVEs with CVSS scores above 7 compared with project's Maturity Level
CVEs present on the system with CVSS score above 7
Analysis Model
Algorithm Combining Measures and Decision Criteria
EAL Rating
Derived Measure
Derived Measure
Two or More Measures
CMMI Maturity Level
Number of or CWEs per set number of lines of code
Measurement Function
Algorithm Combining Two or More Base Measures
CVSS Score
Measurement Process
Base Measure
Base Measure
A Measure of a Single Attribute
Number of lines of code
Number of CVEs or CWEs
Measurement Method
Measurement Method
Operations Quantifying an Attribute Against a Scale
MOF Element
Measured Artifact
Measurement
Attribute
Attribute
Entities
Property Relevant to Information Needs
Line of code
CVE/CWE/defect
Adapted from ISO/IEC 15939 - Software Measurement Process

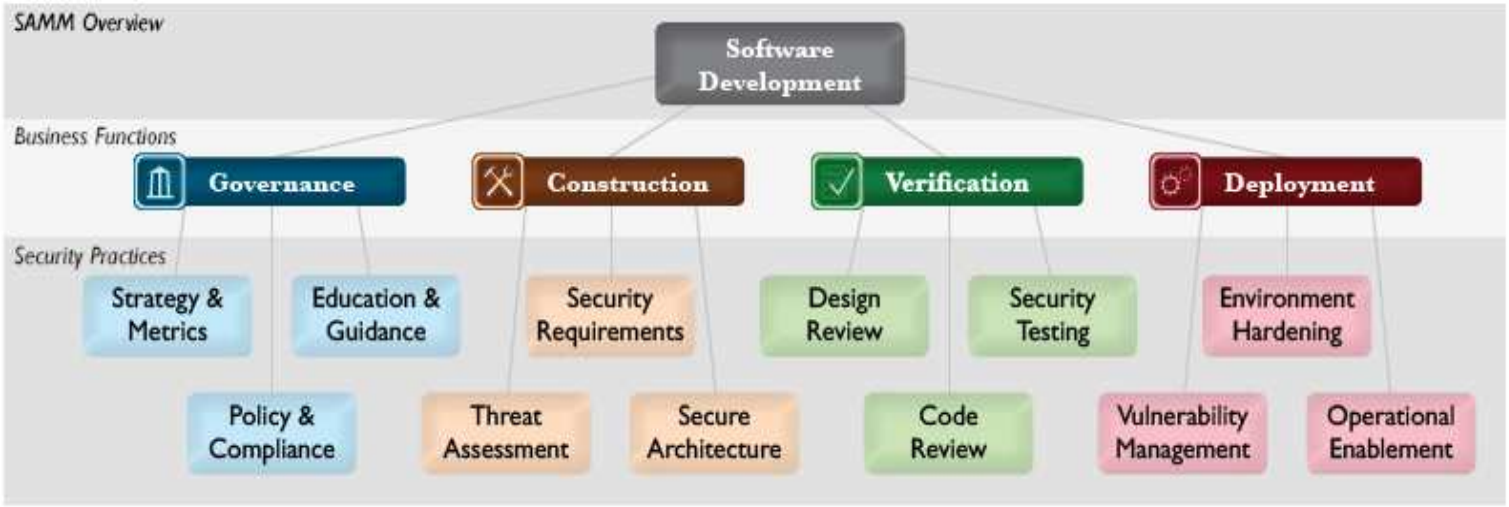
- Identify shortfalls in security knowledge of in-house programmers and help those individuals close the gaps.
- Ensure outsourced programmers have adequate secure coding skills.
- Select new employees who will not need remedial training in secure programming.
- Ensure each major development project has at least one person with advanced secure programming skills.

http://www.sans-ssi.org/certification/

- The Problem
  - Security is not being addressed from a holistic perspective throughout the software lifecycle. Some 80% of all security breaches are application related. Every person involved should consider security as an essential element.
- The Solution
  - Professional Certification – with CSSLPCM, we will establish an industry standard and instill best practices.

http://www.isc2.org/uploadedFiles/(ISC)2_Public_Content/Certification_Programs/CSSLP/CSSLP-Brochure-ForPDF.pdf

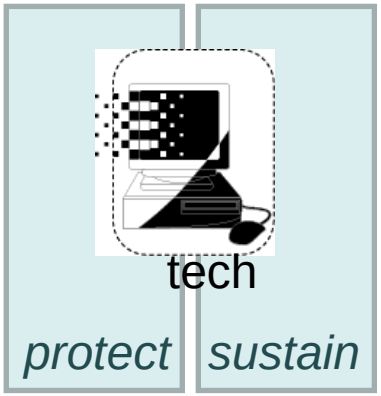
- **Open Software Assurance Maturity Model (SAMM)**
  - http://www.opensamm.org/
  - Open framework to help organizations formulate and implement a strategy for software security tailored to specific risks



http://www.opensamm.org/downloads/SAMM-1.0.pdf

- **Building Security In Maturity Model (BSIMM)**
  - http://www.bsimm2.com/
  - Is designed to help understand and plan a software security initiative
  - BSIMM was created through a process of understanding and analyzing real-world data from nine leading software security initiatives
  - BSIMM uses a Software Security Framework (SSF), to provide a conceptual scaffolding for the model
  - Properly used, BSIMM can help determine where your organization stands with respect to real-world software security initiatives and what steps can be taken to make your approach more effective.
- **BSIMM**
  - Not a complete "how to" guide for software security, nor is it a one size fits all model
  - It is a collection of good ideas and activities that are in use today

- Resiliency requirements form basis for protection and sustainment of an asset
- Resiliency requirements are informed by
  - Organization's mission and strategy
  - Role of the asset in the service
  - Asset interdependencies
- Resiliency requirements must be addressed in development & acquisition of new software assets



CERT® Resiliency Management Model (RMM) is a process improvement model that addresses
Convergence of security, business continuity, and IT operations to manage operational Risk and establish operational resiliency

http://www.cert.org/resiliency/rmm.html

Source: Evolution in Software Assurance Processes Panel, David White, SwA Forum November 2009

Source: CERT® Resiliency Management Model, Lisa Young, SwA Working Groups December 2009

- Overview Of Challenges In The Implementation Of SwA Practices
- Understanding Practice Implementation (A Self Assessment Approach)
- Leveraging The Practice Implementation Self Assessment During Acquisition

- Why do developers reuse untested code without determining if it is "fit-for-purpose"?
- Why do organizations acquire code from various unknown suppliers with unknown levels of assurance?
- Why are acquirers unaware of how to assess and compare vendors' software assurance and supply chain risk management activities?
- Why is software continuously exploited?

- 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

- Assurance Process Reference Model for CMMI (PRM)
- Open Software Assurance Maturity Model (OSAMM)
- Building Security In Maturity Model (BSIMM)
- Resiliency Management Model (RMM)
- Capability Maturity Model Integration for Acquisition (CMMI–ACQ)

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**
GOALS:	Establishes and executes plan for ensuring software is secured throughout the supply chain	Enforces and tracks compliance with security plan policies and other compliance requirements	Fosters training and awareness programs to ensure staff can properly maintain a secure software supply chain	Performs threat modeling and maintains knowledgebase of threats to secure software supply chain	Develops and enforces security requirements that will ensure a secure software supply chain	Builds security into the software design	Reviews software designs to ensure they meet the documented assurance requirements	Analyzes code to mitigate bugs before advancing to production	Performs automated testing as part of QA process to identify flaws	Conducts penetration testing to test software from a hacker's perspective	Establishes robust processes to identify, prioritize, and fix software vulnerabilities	Protects, monitors, and manages the software environment	Manages supplier risk and documents supplier security requirements	Reviews and selects supplier(s) demonstrating sufficient risk management controls and processes to meet security requirements	Enforces, monitors, manages, and analyzes supplier performance against documented supplier security requirements
Practices:	Establisher Security Plan; communicater and provider training for the plan	Identifier and maintainer relevant compliance drivers	Conducter security awareness training regularly	Builder and maintainer list of application-specific attack models	Documenter, analyzer, and manager functional security requirements	Developer list of preferred frameworks and security features; explicitly applier security principles to design	Reviewer design against security requirements	Developer list of top bugs and greater review checklists from security requirements	Performer edge/boundary value condition testing in QA process	Performer external penetration testing and production software with latest techniques and mitigater defects	Identifier point of contact for incident response; creater incident response team	Maintainer operational environment specification	Determiner and prioritizer supplier dependencies; identifier, assessor, and mitigater risks associated with	Establisher, reviewer, and distributer solicitation package	Formalizer supplier relationships and executer supplier agreements
Status:															
Practices:	Collecter and tracker Security Plan metrics based upon risk	Establisher solicitor and procedurer for compliance with security plan and other compliance requirements	Conducter role-based advanced application security training	Identifier potential attacker profiles	Documenter, analyzer, and manager non-functional security requirements	Builder secure framework, security reviewer, and security design patterns	Maker design reviewer available for projects	User automated code analysis tools; requirer code analysis as part of development process	Integrater black box security testing tools into QA of software releases	Performer periodic internal white box testing	Developer consistent incident response process	Monitorer baseline environment configuration changes	Establisher enterprise and assurance requirements for supplier agreements	Evaluater solicitation responses	Monitorer and corrector supplier processes and performance
Status:															
Practices:	Driver budgets based upon analysis from metrics collections	Measurer project compliance at specific checkpoints	Provider security refresher for coaching / learning	Builder and maintainer abuse cases and attack patterns	Builder knowledgebase of well-written, reusable, testable security requirements	Requirer use of approved security platforms and architectures	Builder standard architectural patterns from lessons learned	Tailorer code analysis for application-specific concerns	Employer risk-driven automated security and regression testing in QA process	Performer extensive penetration testing customized with organizational knowledge	Conducter root cause analysis for incidents, fixing all occurrences of bugs	Identifier and deployer relevant operating and protection tools; performer code signing	Establisher supplier agreements	Negotiator and selector supplier	Evaluator and assessor supplier work products
Status:															

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**
Practices:	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
BSIMM	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	-	-
	-	CP1.2	T3.4	AM1.4	-	SFD1.2	SFD3.1	-	-	-	-	SE1.2	-	-	-
CMMI-ACQ	PP SG2 - SG3	OPF SG1	OT SG2	RSKM SG1 - SG2	ARD SG1, SG3	ATM SG2	ATM SG1	AVER SG3	AVER SG3	AVER SG3	CAR SG1	CM SG2 - SG3	RSKM SG2-SG3	SSAD SG1	AM SG1
	-	-	-	-	REQM SG1	AVAL SG2	AVAL SG1 - SG2	-	-	CAR SG1 - SG2	OPD SG1	-	PP SG1	-	SSAD SG3
OSAMM	SM1B	PC1A	EG1A	TA1A	SR1A	SA1A	DR1B	CR1A	ST2B	ST1B	VM1A	EH1A	-	-	-
	-	PC1B	-	-	SR2B	SA1B	-	-	-	-	VM1B	-	-	-	-
PRM	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 1.3	-	-	-	-	-	-	-	-	-	-	-	SG 3.1	-	-
RMM	RTSE:SG2 - SG3	COMP:SG2	OTA:SG1 - SG2	RISK:SG1 - SG4	RRD:SG1 - SG3	RTSE:SG1 - SG2	-	VAR:SG2	RTSE:SG3	RTSE:SG3	VAR:SG1	ADM:SG3	EXD:SG1 - SG2	EXD:SG3	EXD:SG3
	MON:SG1	MON:SG1 - SG2	-	KIM:SG6	RRM:SG1	KIM:SG2, SG6	-	KIM:SG6	-	-	MON:SG1	KIM:SG5	RISK:SG3 - SG6	-	-
Practices:	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
BSIMM	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	-	-
	SM2.1	CP3.2	-	-	-	SFD2.3	AA2.3	CR2.3	-	-	-	-	-	-	-
CMMI-ACQ	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
	PMC SG1	-	-		REQM SG1	AVAL SG2	PMC SG1 - SG2	-	-	-	OPD SG1	-	ARD SG2	-	REQM SG1
OSAMM	SM1B	PC2A	EG2A	TA1B	SR1B	SA2A	DR2A	CR2A	ST1B	ST1A	VM2A	EH2B	SR3A	-	-
	-	-	EG3B	-	-	SA2B	DR2B	CR2B	-	ST1B	-	-	-	-	-
PRM	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 2.2	-	-	-	-	-	-	-	-	-	-	-	-	-	SG 3.5
RMM	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:SG4
	MON:SG2	COMP:SG1	-	KIM:SG6	RRM:SG1	-	-	-	-	-	MON:SG1	KIM:SG5	RRD:SG2 - SG3	-	RRM:SG1
Practices:	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
BSIMM	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	-	-
	-	CP3.3	T2.4 - T2.5	AM 2.2	SR2.3	-	-	-	-	-	-	-	CP3.2	-	-
CMMI-ACQ	PMC SG2	OPP 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
OSAMM	SM3A	PC3A	EG1B - EG2B	TA2A	SR2A	SA3A	DR3A	CR3A	ST1A	ST1B	VM3A	EH3A	-	-	-
	SM3B	-	EG3A	-	-	SA3B	-	-	ST2A	-	-	OE3B	-	-	-
PRM	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 3.5	-	-	-	-
RMM	RTSE:SG3.SP1	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
	MON:SG2	COMP:SG3 - SG4	OTA:SG4	KIM:SG6	-	-	-	RTSE:SG3	-	-	MON:SG2	-	-	-	RRM:SG1

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

Source: SwA Benchmarking and Implementation, Moss, SSTC 2010

Assurance PRM	SAFEcode	MS SDL	Open SAMM	BSIMM
•Identify deviations from assurance coding standards •Ensure adequate resources	•Fundamental Practices for Secure SW Development (section on Programming)	1. Basic code scanning tools 2. Evaluate and recommend appropriate security tools 3. Use of static analysis tools 4. In-house security tool customization	1. Create review checklists from known security requirements 2. Utilize automated code analysis tools 3. Customize code analysis for application specific concerns	1. Provide easily accessible security standards and (compliance-driven) requirements 2. Enforce standards through mandatory automated code review and centralized reporting 3. Build an automated code review factory with tailored rule

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

SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

SwA Self-Assessment (High Level)

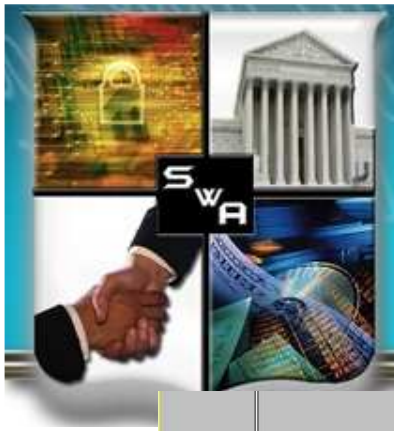
Role	Goal	Expected Practice	Activities	Source	BSIMM	CMMI-ACQ	OSAM M	RMM	MS SDL	Developer Considerations	Acquirer Considerations	Practice Implementation Level	Notes
DEV	SG 3.1 Establish assurance requirements.	SP 3.1.1 Understand the operating environment and define the operating constraints for assurance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

SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

SwA Self-Assessment (Mappings)

Role	Goal	Expected Practice	Activities	Source	BSIMM	CMMI-ACQ	OSAMM	RMM	MS SDL
DEV	SG 3.1 Establish assurance requirements.	SP 3.1.1 Understand the operating environment and define the operating constraints for assurance within the environments of system deployment.	Identify the system assurance context. Identify the system vulnerabilities with each operating environment defined for the system. Identify applicable assurance laws, policies, and constraints.	AF RD SP 1.1		PP SG1	EH1A	EF SG1 - SG2	
		SP 3.1.2 Develop customer assurance requirements.		AF RD SP 1.2	SR1.1	ARD SG1, SG3	SR1A	RRD:SG1 - SG3	
					SR1.2	REQM SG1	SR1B	COMP:SG2	
					SR1.3		SR2A	KIM:SG6	
					SR2.3		SR2B	RRM:SG1	
		SP 3.1.3 Define product and product component assurance requirements		AF SP 2.1	SFD3.2	CM SG1	SA3A	KIM:SG2	P7
							SA3B		P2
		SP 3.1.4 Identify operational concepts and associated scenarios for intended and unintended use and associated assurance considerations.		AF RD SP3.1	AM1.1	RSKM SG1 - SG2	TA1A	RISK:SG1 - SG4	
					AM1.3		TA1B	KIM:SG6	
					AM1.4		TA2A		
					AM2.1				
					AM2.2				
		SP 3.1.5 Analyze assurance requirements.	Ensure established assurance requirements for the product flow to lower level solutions. Verify requirements against assurance objectives	AF RD SP 3.5	SR1.3	ARD SG3	SR1B	RRD:SG3	
		SP 3.1.6 Balance assurance needs against cost benefits.		AF SP 3.4	SM1.5	ARD SG3	SM3A - SM3B	FRM:SG4 - SG5, RRD:SG3	
		SP 3.1.7 Obtain Agreement of risk for Assurance level.			SM2.4	RSKM SG2	SM1A	RISK SG4, KIM SG3	
DEV		SP 3.2.1 Develop alternative solutions and	Identify assurance defects and effectiveness of corrective actions in relevant products/systems/operations and apply	TS SP 1.1	SFD1.1	ATM SG2	SA1A	RTSE:SG1 - SG2	



SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

SwA Self-Assessment (Considerations)

Role	Goal	BSIMM	CMMI-ACQ	OSAMM	RMM	MS SDL	Developer Considerations	Acquirer Considerations	Practice Implementation Level	Notes
DEV			PP SG1	EH1A	EF SG1 - SG2					
		SR1.1	ARD SG1, SG3	SR1A	RRD:SG1 - SG3					
		SR1.2	REQM SG1	SR1B	COMP:SG2					
		SR1.3		SR2A	KIM:SG6					
		SR2.3		SR2B	RRM:SG1					
		SFD3.2	CM SG1	SA3A	KIM:SG2	P7				
				SA3B		P2				
	SG 3.1 Establish assurance requirements.	AM1.1	RSKM SG1 - SG2	TA1A	RISK:SG1 - SG4					
		AM1.3		TA1B	KIM:SG6					
		AM1.4		TA2A						
		AM2.1								
		AM2.2								
		SR1.3	ARD SG3	SR1B	RRD:SG3					
		SM1.5	ARD SG3	SM3A - SM3B	FRM:SG4 - SG5, RRD:SG3					
		SM2.4	RSKM SG2	SM1A	RISK SG4, KIM SG3					
DEV		SFD1.1	ATM SG2	SA1A	RTSE:SG1 - SG2					

- Overview Of Challenges In The Implementation Of SwA Practices
- Understanding Practice Implementation (A Self Assessment Approach)
- Leveraging The Practice Implementation Self Assessment During Acquisition

- 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

What should we consider from the acquisition community's perspective as we move forward?