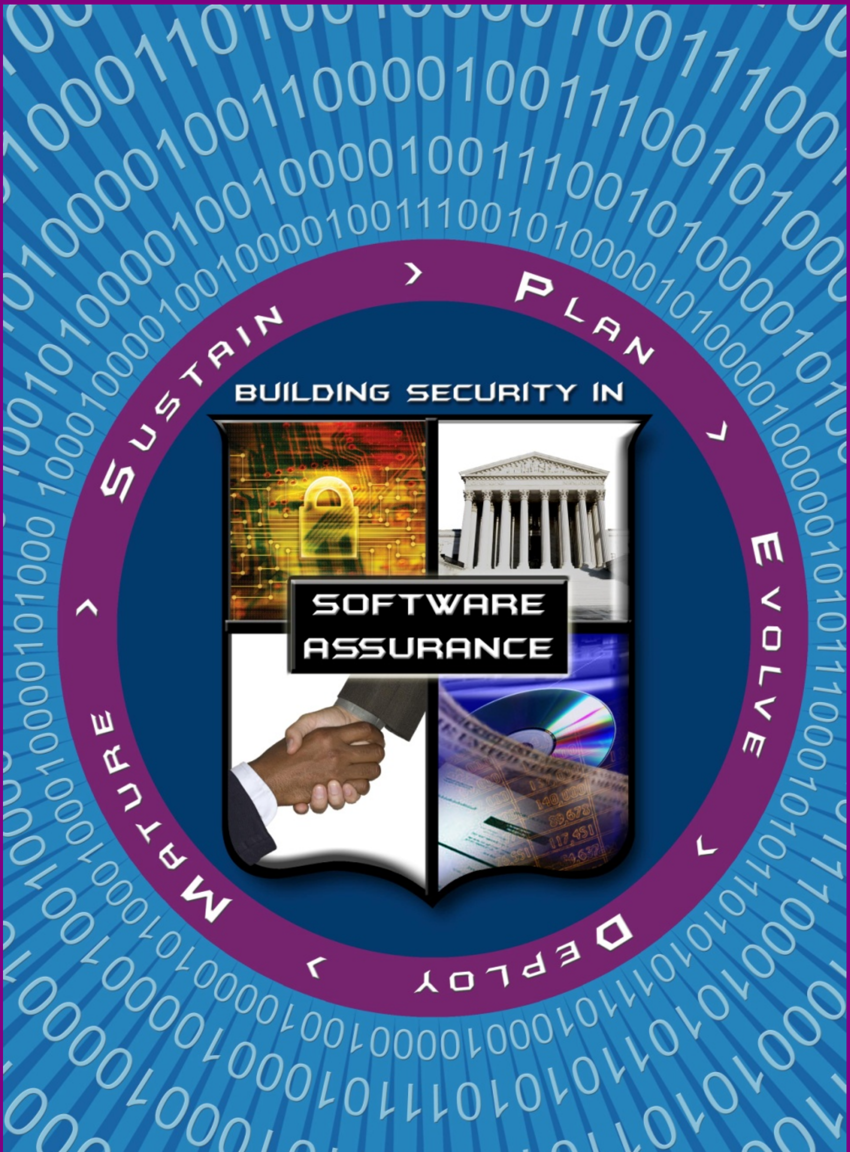
Software Assurance in Education, Training & Certification

Software Assurance Pocket Guide Series:
Life Cycle Support, Volume I
Version 1.0, May 28, 2010 (Draft)



Software Assurance (SwA) Pocket Guide Resources

This is a resource for ‘getting started’ in educating, training and certifying a workforce with regards to their awareness about the engineering activities and knowledge areas in building software that is free from vulnerabilities, either intentionally designed into the software or accidentally inserted at any time during its life cycle, and that the software operates as expected. As part of the Software Assurance (SwA) Pocket Guide series, this resource is offered for informative use only; it is not intended as directive or presented as being comprehensive since it references and summarizes material in the source documents that provide detailed information. When referencing any part of this document, please provide proper attribution and reference the source documents, when applicable.

This volume of the SwA Pocket Guide series focuses on enumerating education, training and certification resources. It identifies the most effective strategies to inject software assurance topics into existing college curriculums and workforce training and certification programs.

At the back of this pocket guide are references, limitation statements, and a listing of topics addressed in the SwA Pocket Guide series. All SwA Pocket Guides and SwA-related documents are freely available for download via the SwA Community Resources and Information Clearinghouse at https://buildsecurityin.us-cert.gov/swa.

BUILDING SECURITY IN

SOFTWARE ASSURANCE

Acknowledgements

SwA Pocket Guides are developed and socialized by the SwA community as a collaborative effort to obtain a common look and feel and are not produced by individual entities. SwA Forum and Working Groups function as a stakeholder meta-community that welcomes additional participation in advancing software security and refining. All SwA-related information resources that are produced are offered free for public use. Inputs to documents for the online resources are invited. Please contact Software.Assurance@dhs.gov for comments and inquiries. For the most up to date pocket guides, check the website at https://buildsecurityin.us-cert.gov/swa/.

The SwA Forum and Working Groups are composed of government, industry, and academic members and focuses on incorporating SwA considerations in the acquisition and development processes relative to potential risk exposures that could be introduced by software and the supply chain.

Participants in the SwA Forum’s Workforce Education and Training Working Group contributed to developing the material used in this pocket guide as a step in raising awareness on how to incorporate SwA topics in education,

training and certification of a workforce that is knowledgeable to perform engineering activities or aspects of activities relevant for promoting software assurance throughout the Software Development Life Cycle (SDLC).

Information contained in this pocket guide is primarily derived from the documents listed in the *Resource* boxes that follow throughout this pocket guide.

Special thanks go to Assistant Professor, Robin A. Gandhi, Ph.D., at the University of Nebraska at Omaha, for providing the synthesis and organization of the material, as well as and the Department of Homeland Security (DHS), National Cyber Security Division's Software Assurance team who provided much of the support to enable the successful completion of this guide and related SwA documents. We also acknowledge reviews, contributions, and several discussions by members of the SwA Forum's Workforce Education and Training Working Group to improve this document.

Resources

"Software Assurance: A Curriculum Guide to the Common Body of Knowledge", DHS SwA Forum Workforce Education and Training Working Group, Samuel T. Redwine, Jr. (Editor), Version 1.2, U.S. Department of Homeland Security (DHS), October 2007 at https://buildsecurityin.us-cert.gov/daisy/bsi/940-BSI/version/default/part/AttachmentData/data/CurriculumGuideToTheCBK.pdf.

"Software Security Assurance: A State-of-the-Art Report", Goertzel, Karen Mercedes, *et al*, Information Assurance Technology Analysis Center (IATAC) of the Defense Technical Information Center (DTIC) at http://iac.dtic.mil/iatac/reports.jsp.

"Enhancing the Development Life Cycle to Produce Secure Software: A Reference Guidebook on Software Assurance," Goertzel, Karen, Theodore Winograd, et al. for Department of Homeland Security and Department of Defense Data and Analysis Center for Software., October 2008 at https://www.thedacs.com/techs/enhanced_life_cycles/.

NASA Software Assurance Guidebook, at http://sato.gsfc.nasa.gov/guidebook/index.php.

"IT Security Essential Body of Knowledge (EBK): A Competency and Functional Framework for IT Security Workforce Development," DHS U.S. Computer Emergency Response Team (US-CERT), at http://www.us-cert.gov/ITSecurityEBK/.

DoD 8570.01-M, "Information Assurance Workforce Improvement Program," Incorporating Change 2, April 20, 2010, Assistant Secretary of Defense for Networks and Information Integration/Department of Defense Chief Information Officer, at http://www.dtic.mil/whs/directives/corres/pdf/857001m.pdf.

"Towards an Organization for Software System Security Principles and Guidelines," version 1.0, Samuel T. Redwine, Jr.,.Institute for Infrastructure and Information Assurance, James Madison University, IIIA Technical Paper 08-01. February 2008 at http://www.jmu.edu/iiia/webdocs/Reports/SwA_Principles_Organization-sm.pdf.

"Integrating Software Assurance Knowledge Into Conventional Curricula" Crosstalk: The Journal of Defense Software Engineering, Jan 2008, Mead, N.R., Shoemaker, D., & Ingalsbe, J.A., at http://www.stsc.hill.af.mil/crossTalk/2008/01/0801MeadShoemakerIngalsbe.html.

Software Assurance Curriculum Project Volume I: Master of Software Assurance Reference Curriculum. Mead, Nancy R.; Allen, Julia H.; Ardis, Mark; Hilburn, Thomas B.; Kornecki, Andrew J.; Linger, Rick; & McDonald, James. (CMU/SEI-2010-TR-005, ESC-TR-2010-005). Software Engineering Institute, Carnegie Mellon University, 2010. http://www.sei.cmu.edu/library/abstracts/reports/10tr005.cfm

Software Assurance Curriculum Project Volume II: Undergraduate Course Outlines. Mead, Nancy R.; Hilburn, Thomas B.; & Linger, Rick. Software Assurance Curriculum Project Volume II: Undergraduate Course Outlines (CMU/SEI-2010-TR-019, ESC-TR-2010-019). Software Engineering Institute, Carnegie Mellon University, 2010. http://www.sei.cmu.edu/library/abstracts/reports/10tr019.cfm

Overview

Current events related to cybersecurity encourage a fundamental shift in the way we think about educating and training a workforce prepared to address security issues in all phases of a software system. Software assurance education and training is aimed to ensure adequate coverage of requisite knowledge areas in contributing disciplines such as software engineering (including its many subdisciplines), systems engineering, project management, etc., to identify and acquire competencies associated with secure software. The primary audiences for this pocket guide are educators and trainers who can use this guide to identify resources to supplement their efforts as well as identify strategies to inject software assurance related topics in the existing education and training programs.

Guiding Questions for SwA Curriculum Development:

Activities: What are the engineering activities or aspects of activities that are relevant to achieving secure software?

Knowledge: What knowledge is needed to perform these activities or aspects?

The objective of software assurance is to ensure that the processes, procedures, and products used to produce and sustain the software conform to all requirements and standards specified to govern those processes, procedures, and products. Software assurance in its broader sense refers to the assurance of any required property of software. However, in the context of this pocket guide, software assurance is concerned with assuring the security of software.

Building secure software requires a workforce that understands the processes and technologies necessary to provide the basis for belief that software will consistently exhibit all properties required to ensure that the software will operate as expected; despite the presence of faults introduced by a malicious adversary. The Ware Report (1969) identified that:

> "*Probably the most serious risk in system software is* ***incomplete design****, in the sense that inadvertent loopholes exist in the protective barriers and have not been foreseen by the designers.*"

Later the Anderson Report (1972) clearly established the technical problem to be solved as that of:

> "…*determining what constitutes an appropriate defense against malicious attack, and then developing hardware and software with the defensive mechanisms* ***built in***."

Nearly forty years after, as we find ourselves in the midst of a highly interconnected cyber infrastructure the need for a workforce with better skills to **build security in** cannot be emphasized enough. The objective is to enable a workforce competent in managing, designing, implementing and evaluating systems that can enforce security policies and fulfill security expectations. This workforce should be able to develop a well-reasoned and auditable basis for believing that the software will function as expected, i.e. have justifiable arguments to questions such as:

» How secure is your software?
» What is it secure against?
» How does it achieve its security goals?

The Case for Software Assurance Education

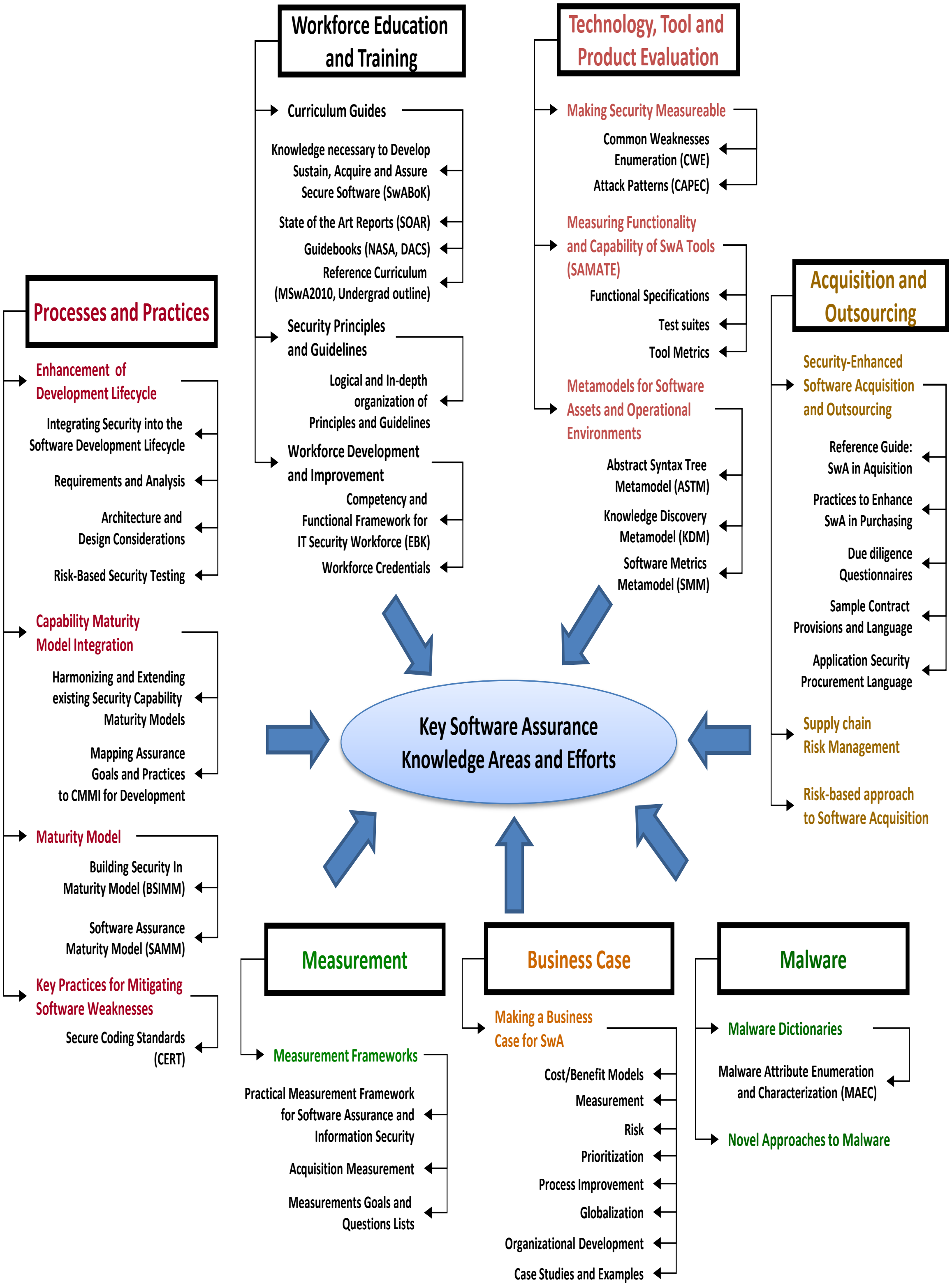
Software assurance has become critical because dramatic increases in business and mission risks are now known to be attributable to exploitable software: system interdependence and software dependence has software as the weakest link; software size and complexity obscures intent and precludes exhaustive test; outsourcing and use of un-vetted software supply chain increases risk exposure; attack sophistication eases exploitation; reuse of legacy software interfaced with other applications in new environments introduces other unintended consequences increasing the number of vulnerable targets; and the number of threats targeting software. These all contribute to the increase of risks to software-enabled capabilities and the threat of asymmetric attack. A broad range of stakeholders now need confidence that the software which enables their core business operations can be trusted to perform (even with attempted exploitation).

In their report to the President, Cyber Security: A Crisis of Prioritization (February 2005), in the chapter entitled "Software Is a Major Vulnerability", the President's Information Technology Advisory Committee (PITAC) summed up the problem of non-secure software concisely and accurately:

> "*Network connectivity provides "door-to-door" transportation for attackers, but vulnerabilities in the software residing in computers substantially compound the cyber security problem. As the PITAC noted in a 1999 report, the software development methods that have been the norm fail to provide the high quality, reliable, and secure software that the Information Technology infrastructure requires.*
>
> *Software development is not yet a science or a rigorous discipline, and the development process by and large is not controlled to minimize the vulnerabilities that attackers exploit. Today, as with cancer, vulnerable software can be invaded and modified to cause damage to previously healthy software, and infected software can replicate itself and be carried across networks to cause damage in other systems. Like cancer, these damaging processes may be invisible to the lay person even though experts recognize that their threat is growing. And as in cancer, both preventive actions and research are critical, the former to minimize damage today and the latter to establish a foundation of knowledge and capabilities that will assist the cyber security professionals of tomorrow reduce risk and minimize damage for the long term.*
>
> *Vulnerabilities in software that are introduced by mistake or poor practices are a serious problem today. In the future, the Nation may face an even more challenging problem as adversaries - both foreign and domestic – become increasingly sophisticated in their ability to insert malicious code into critical software.*"

It is clear that to produce, acquire, and sustain secure software, a framework that identifies workforce needs for competencies, leverages sound practices, and guide curriculum development for education and training relevant to software assurance is inevitable. Because software quality assurance and software engineering have evolved bodies of knowledge that do not explicitly address security as a quality attribute, a workforce education and training framework must also identify the integration point of secure software development techniques and practices in the existing programs nationwide.

Key SwA Knowledge Areas and Efforts



Key Software Assurance Knowledge Areas and Efforts

Workforce Education and Training

- Curriculum Guides
  - Knowledge necessary to Develop Sustain, Acquire and Assure Secure Software (SwABoK)
  - State of the Art Reports (SOAR)
  - Guidebooks (NASA, DACS)
  - Reference Curriculum (MSwA2010, Undergrad outline)
- Security Principles and Guidelines
  - Logical and In-depth organization of Principles and Guidelines
- Workforce Development and Improvement
  - Competency and Functional Framework for IT Security Workforce (EBK)
  - Workforce Credentials

Technology, Tool and Product Evaluation

- Making Security Measureable
  - Common Weaknesses Enumeration (CWE)
  - Attack Patterns (CAPEC)
- Measuring Functionality and Capability of SwA Tools (SAMATE)
  - Functional Specifications
  - Test suites
  - Tool Metrics
- Metamodels for Software Assets and Operational Environments
  - Abstract Syntax Tree Metamodel (ASTM)
  - Knowledge Discovery Metamodel (KDM)
  - Software Metrics Metamodel (SMM)

Processes and Practices

- Enhancement of Development Lifecycle
  - Integrating Security into the Software Development Lifecycle
  - Requirements and Analysis
  - Architecture and Design Considerations
  - Risk-Based Security Testing
- Capability Maturity Model Integration
  - Harmonizing and Extending existing Security Capability Maturity Models
  - Mapping Assurance Goals and Practices to CMMI for Development
- Maturity Model
  - Building Security In Maturity Model (BSIMM)
  - Software Assurance Maturity Model (SAMM)
- Key Practices for Mitigating Software Weaknesses
  - Secure Coding Standards (CERT)

Acquisition and Outsourcing

- Security-Enhanced Software Acquisition and Outsourcing
  - Reference Guide: SwA in Aquisition
  - Practices to Enhance SwA in Purchasing
  - Due diligence Questionnaires
  - Sample Contract Provisions and Language
  - Application Security Procurement Language
- Supply chain Risk Management
- Risk-based approach to Software Acquisition

Measurement

- Measurement Frameworks
  - Practical Measurement Framework for Software Assurance and Information Security
  - Acquisition Measurement
  - Measurements Goals and Questions Lists

Business Case

- Making a Business Case for SwA
  - Cost/Benefit Models
  - Measurement
  - Risk
  - Prioritization
  - Process Improvement
  - Globalization
  - Organizational Development
  - Case Studies and Examples

Malware

- Malware Dictionaries
  - Malware Attribute Enumeration and Characterization (MAEC)
- Novel Approaches to Malware

SwA Curriculum and Training Development Guides

Table 1– SwA Curriculum and Training Development Guides

Identifier	Relevant Documents and Links	Purpose
• SwA Curriculum Project	Volume I: Master of Software Assurance Reference Curriculum. Mead, Nancy R. et al. SEI/CMU. http://www.cert.org/mswa/ ; http://www.cert.org/podcast/show/20101026mead.html ;	Offers a core body of knowledge from which to create a master's level degree program in software assurance, as a standalone offering and as a track within existing software engineering and computer science master's degree programs. Last updated **2010.**
	Volume II: Undergraduate Course Outlines. Mead, Nancy R. et al. SEI/CMU. http://www.cert.org/mswa/	Focuses on an undergraduate curriculum specialization for software assurance. Intended to provide students with fundamental skills for either entering the field directly or continuing with graduate level education. Last updated **2010.**
• Software Security Assurance SOAR	Software Security Assurance: A State-of-the-Art Report. Goertzel, Karen Mercedes, et al, IATAC of the DTIC. http://iac.dtic.mil/iatac/download/security.pdf	Identifies the current "state-of-the-art" in software security assurance. Last updated July **2007**.
	Enhancing the Development Life Cycle to Produce Secure Software: A Reference Guidebook on Software Assurance. Goertzel, Karen et al. For DHS and DTIC https://www.thedacs.com/techs/enhanced_life_cycles/	Complements the Software Security Assurance: A State-of-the-Art Report with further details. Last updated October **2008**
• SwA CBK	Software Assurance Body of Knowledge. Version 1.2, Samuel T. Redwine, Jr. (Editor), DHS, https://buildsecurityin.us-cert.gov/bsi/dhs/927-BSI.html	A comprehensive set of principles and guidelines from the disciplines of software engineering, systems engineering, information system, computer science, safety, security, testing, information assurance, and project management. Last updated October **2007**.
	Towards an Organization for Software System Security Principles and Guidelines. Version 1.0, Samuel T. Redwine, Jr, https://buildsecurityin.us-cert.gov/bsi/dhs/927-BSI.html	An extensive set of software system security principles and guidelines organized in a logical, in-depth fashion. Last updated February **2008**.

Workforce Development and Improvement

Table 2– Workforce Development and Improvement

Identifier	Relevant Documents and Links	Purpose
• **EBK**	IT Security Essential Body of Knowledge (EBK): A Competency and Functional Framework for IT Security Workforce Development. DHS US-CERT http://www.us-cert.gov/ITSecurityEBK/	Characterizes the IT security workforce and provides a national baseline representing the essential knowledge and skills that IT security practitioners should have to perform specific roles and responsibilities. Last updated September **2008.**
• **DoD 8570.01-M**	Information Assurance Workforce Improvement Program. Assistant Secretary of Defense for Networks and Information Integration/Department of Defense Chief Information Officer. http://www.dtic.mil/whs/directives/corres/pdf/857001m.pdf	Provides guidance and procedures for the training, certification, and management of the DoD workforce conducting Information Assurance (IA) functions in assigned duty positions. Last update: Incorporating Change 2, April 20, **2010**.

Strategies for Injecting SwA Knowledge Areas in existing Education and Training Programs

Table 3– Strategies

Strategy	Relevant Documents and Links
• **Degree programs and specializations in SwA**	Reference curriculums available from the Software Engineering Institute, Carnegie Mellon University can be used as recommendations for designing Masters of Software Assurance degree program and undergraduate curriculum specialization in software assurance. These reference curriculum are available at http://repository.cmu.edu/sei/3/ and http://repository.cmu.edu/sei/4/
• **Stand-alone Courses**	New course offerings based on SwA knowledge areas complement existing Software Engineering courses. Examples: http://www.cs.jmu.edu/sss https://www.securecoding.cert.org/confluence/display/sci/S08+15392+Secure+Programming Other examples include graduate-level Software Assurance courses that cover the secure software engineering activities during the SDLC, being offered at the University of North Carolina at Charlotte, and The University of Nebraska at Omaha
• **Augmenting Existing Courses**	The SwA CBK and State-of-the-Art reports are catalogs of secure software development practices, processes, and techniques that can be mapped to topics relevant to current curriculums. The identified gaps can then be filled

	using relevant materials.
• **Micro-Modules**	Problem-based learning exercises, in class workshops or short talks can be conducted to inject topics such as Mis-use cases and Assurance Cases in existing software engineering or information security courses.
• **Capstone and Class Projects**	Software Engineering capstone courses or class projects can be geared towards a security critical domain such as designing a software system for the Department of Defense, Cyber-physical systems or for a Credit Card transaction processing company. These domains will facilitate the exploration of security needs throughout the SDLC.
• **Online Courses**	The Adaptive Cyber-Security Training Online (**ACT-Online**) courses are available on the TEEX Domestic Preparedness Campus. Ten courses are offered through three discipline specific tracks targeting everyday non-technical computer users, technical IT professionals, and business managers and professionals. These courses are offered at no cost and students earn a DHS/FEMA Certificate of completion along with Continuing Education Units (CEU) at the completion of each course. http://www.teexwmdcampus.com/index.k2
	The **CERT Virtual Training Environment (VTE)** combines the components of traditional classroom training with the convenience of web-based training. Over 200 hours of course material focused around the technical, policy, and management implications of information security – including preparatory courses for commercial certifications, core skills courses, role-based courses for managers and technical staff, and vendor-developed courses. Open access is provided to individual DoD personnel (Active Duty, DoD Civilian and contractors) and members of the Federal Civilian Workforce through specific sponsorships from DISA, and DHS in conjunction with the Department of State Foreign Service Institute. Sponsored accounts can be requested at www.vte.cert.org. Public access to many of the materials is provided through the VTE Library at https://www.vte.cert.org/vteweb/Library/Library.aspx
• **Awareness and Self-study Resources**	**SAFECode**: Software Assurance Forum for Excellence in Code. http://www.safecode.org **Fundamental Practices for Secure Software Development** http://www.safecode.org/publications/SAFECode_Dev_Practices1108.pdf **Security Engineering Training** http://www.safecode.org/publications/SAFECode_Training0409.pdf **Software Assurance: An Overview of Current Industry Best Practices** http://www.safecode.org/publications/SAFECode_BestPractices0208.pdf **Framework for Software Supply Chain Integrity** http://www.safecode.org/publications/SAFECode_Supply_Chain0709.pdf **Software Integrity Controls: An Assurance-Based Approach to Minimizing Risks in the Software Supply Chain.** http://www.safecode.org/publications/SAFECode_Software_Integrity_Controls0610.pdf
	Rugged Software http://www.ruggedsoftware.org/

• **Community Support**	**Linkedin SwA Education Discussion Group** Nancy Mead, SwA Curriculum Team lead The objective of the SwA Curriculum Development Team in establishing this group is to provide a venue for dialog about software assurance education. http://www.linkedin.com/groups?mostPopular=&gid=3430456

SwA Tools in Education and Training

Tools and web resources that can be used in class to provide hands-on experience with SwA Concepts.

Table 1 – Tools and web resources for hands-on classroom experience with SWA Concepts

Tool Name	Tool Description	Possible Classroom Uses
ArgoUML	ArgoUML is the leading open source UML modeling tool and includes support for all standard UML 1.4 diagrams. It runs on any Java platform.	Mis-use cases, security focused UML class diagrams and other documentation for class assignments and projects.
Microsoft SDL Threat Modeling Tool	The Microsoft SDL Threat Modeling Tool allows software architects to identify and mitigate potential security issues early, when they are relatively easy and cost-effective to resolve. Located at http://www.microsoft.com/security/sdl/getstarted/threatmodeling.aspx	Conduct student group workshops to discuss threats to various design alternatives, while suggesting possible mitigation strategies.
ASCE	ASCE supports the key assurance case notations: Goal Structuring Notation and Claims-Arguments-Evidence. Academic license available upon request at http://www.adelard.com/web/hnav/ASCE/index.html.	Assurance case documentation for class assignments and projects, Demonstration of worked examples used on real projects.
Burp Suite	Burp Suite is an integrated platform for attacking web applications. Located at http://www.portswigger.net/suite/	Burp Suite allows to combine manual and automated techniques to enumerate, analyze, scan, attack and exploit web applications
Pharos	Paros is an open source proxy that traps all HTTP and HTTPS data between server and client, including cookies and form fields, which can be intercepted and modified. Located at http://parosproxy.org/index.shtml	Paros can be used as an introduction to web application security assessment.
CERT Secure Coding Standards	Secure coding standards for commonly used programming languages such as C, C++ and Java. Located at https://www.securecoding.cert.org	Online reference; examples of coding do's and don't's
SDMetrics	Analyze the structural properties of UML models using object-oriented measures of design size, coupling, and complexity. Located at http://www.sdmetrics.com/	Examine object-oriented metrics and measures for design and source code artifacts
Splint	Splint is a tool for statically checking C programs for security vulnerabilities and coding mistakes. Located at http://www.splint.org/	Static analysis code checking activities

FindBugs™	A program which uses static analysis to look for bugs in Java code at http://findbugs.sourceforge.net/.	Scan java code repositories for bugs; Introduction to static code checking activities.
Vine	Provides an intermediate language that x86 code can be translated to for Static analysis. Located at http://bitblaze.cs.berkeley.edu/vine.html	Identify data flows analysis and conduct binary analysis.
Valgrind	Valgrind is an instrumentation framework for building dynamic analysis tools. Located at http://valgrind.org/	Demonstrate dynamic analysis techniques to detect memory management and threading bugs, as well as detailed program profiling.
Olly Debug	OllyDbg is a 32-bit assembly level debugger for Microsoft Windows. Located at www.ollydbg.de/	Emphasize binary code analysis and particularly useful in cases where source is unavailable. Explain Buffer Overflows.
SAMATE Reference Dataset	The purpose of the SAMATE Reference Dataset (SRD) is to provide users, researchers, and software security assurance tool developers with a set of known security flaws. This will allow end users to evaluate tools and tool developers to test their methods. Located at http://samate.nist.gov/index.php/Main_Page.html.	A reference data set can be used in class to reflect upon known flaws in software.
Web Resources		
OWASP Learning Environments	http://www.owasp.org/index.php/Phoenix/Tools	Comprehensive collection of security tools, exploits, vulnerability scanners, defensive tools, application security.
OWASP Web Goat	http://www.owasp.org/index.php/OWASP_WebGoat_Project	WebGoat is a deliberately insecure J2EE web application maintained by OWASP designed to teach web application security lessons.
Google Code University	http://jarlsberg.appspot.com/	Web application exploits and defenses. Topics include cross-site scripting, cross site request forgery, AJAX vulnerabilities, denial of service, etc.
Software Assurance (SwA) Tools Overview	https://buildsecurityin.us-cert.gov/swa/swa_tools.html	A collection of SwA tools inspired by the NIST Software Assurance Metrics And Tool Evaluation (SAMATE) project.

SwA Books for Education and Training

Table 2 – A List of SwA focused Books for Use in Education and Training

Topic	Title and Publisher	Summary and Possible Use
• Software Assurance in SDLC	» **Secure Coding: Principles and Practices**, Mark G. Graff and Kenneth R. van Wyk, O'Reilly, 2003	A practical approach to integrating SwA topics into the SDLC. Great for assignment of additional readings that complement classroom materials. http://www.securecoding.org/
• Information Security	» **Building a Secure Computer System**, Morrie Gasser, 1988	Good reading for Information Security basics.
• Activities to improve SwA during the SDLC	» **Software Security: Building Security In**, Gary McGraw, Addison-Wesley Professional, 2006.	Introduction to Software Security Touchpoints during software development. Possible use a textbook or additional reference material
• Principles and guidelines • Implementation level issues	» **Building Secure Software: How to Avoid Security Problems the Right Way**, John Viega and Gary McGraw, Addison Wesley, 2002	Software Assurance principles and guidelines and Implementation level issues Possible use a textbook or additional reference material
• Attack Patterns • Reverse Engineering • Implementation level issues	» **Exploiting Software: How to Break Code** by Greg Hoglund and Gary McGraw, Addison Wesley, 2004	Understanding attack strategies to build better defenses. Case studies for class discussion http://www.exploitingsoftware.com/
• Design Principles and Techniques	» **High-Assurance Design: Architecting Secure and Reliable Enterprise Applications**, Clifford J. Berg, Addison-Wesley Professional 2005.	Basic principles and techniques that can be applied to the development of business applications.
• Static Analysis	» **Secure Programming with Static Analysis,** Brian Chess, Jacob West, Addison Wesley, 2007.	Detailed discussion of security issues in several open source applications; steps in the static analysis process
• Software Assurance in SDLC	» **Software Security Engineering: A Guide for Project Managers**, Julia Allen, Sean Barnum, Robert J. Ellison, Gary McGraw, Nancy R. Mead, Addison-Wesley, 2008 (ISBN 032150917X).	Software Security Engineering draws extensively on the systematic approach developed for the Build Security In (BSI) Web site. http://www.softwaresecurityengineering.com

Standards of Practice

Table 3– Domain-specific SwA standards used in practice

Standard	Community of practice	Purpose
• **MISRA C**	Motor Industry Software Reliability Association (MISRA). http://www.misra.org.uk/	A software development standard for the C programming language developed by MISRA. Its aims are to facilitate code safety, portability and reliability in the context of embedded systems, specifically those systems programmed in ISO C. There is also a set of guidelines for MISRA C++.
• **The Building Security In Maturity Model (BSIMM2)**	http://bsimm2.com/	Pronounced "bee simm" was created by observing and analyzing real-world data from thirty leading software security initiatives. The BSIMM can help you determine how your organization compares to other real-world software security initiatives and what steps can be taken to make your approach more effective.
• **openSAMM: The Software Assurance Maturity Model**	http://www.opensamm.org/	An open framework to help organizations formulate and implement a strategy for software security that is tailored to the specific risks facing the organization.

Workforce Credentials

Certification and Training Opportunities

Certification Authority	SwA Relevant Certificates	Resources
EC-Council	» EC-Council Certified Secure Programmer (**ECSP**) (Technologies Covered: C/C++, Java, .Net, PHP, SQL)	http://www.eccouncil.org/certification.htm
	» Certified Secure Application Developer (**CSAD**)	
	» Certified Ethical Hacker (**CEH**)	
	» Licensed Penetration Tester (**LPT**)	
GIAC - Global Information Assurance Certification	» GIAC Secure Software Programmer - .NET (**GSSP-NET**)	http://www.giac.org/certifications/
	» GIAC Secure Software Programmer - Java (**GSSP-JAVA**)	
	» GIAC Web Application Penetration Tester	

	(**GWAPT**)	
	» GIAC Certified Penetration Tester (**GPEN**)	
IEEE Computer Society	» Certified Software Development Professional (**CSDP**)	http://www.computer.org/portal/web/certification
ISC^2	» **CSSLP^{CM}** - Certified Secure Software Lifecycle Professional	http://www.isc2.org/csslp-certification.aspx

*Approximate Prices. Please check the respective websites for more details.

Other SwA Education and Training Topics

SANS

According to the Sans Institute, one of the Top 20 coolest careers is **Security-Savvy Software Developer**. The security-savvy software developer leads all developers in the creation of secure software, implementing secure programming techniques that are free from logical design and technical implementation flaws. This expert is ultimately responsible for ensuring customer software is free from vulnerabilities that can be exploited by an attacker.

- **Why It's Cool?**
  - "You get to make something that actually runs and does something (and won't break under pressure)."
  - "These guys are the senior developers by virtue of their programming prowess."
- **How It Makes a Difference?**
  - "No security architecture or policy can compensate for poorly written, buggy, insecure software. If one pays the necessary attention to security when a product is initially developed, one doesn't need to go back and add security later on."
  - "This is where the rubber meets the road. These are the people making a difference where it really matters...in the software that runs the world."
- **How to Be Successful?**
  - The role of security-savvy software developer is challenging and rewarding from multiple perspectives. To be successful, you must understand a multitude of attack vectors used to exploit software to avoid the introduction of flaws. This experience is also needed to leverage the same attack tools and techniques an adversary might use to exploit your software, identifying flaws to be addressed before product shipment. In a development role, your position will be vital to the company's success, including your ability to communicate the techniques used for secure software development to your peers. This can be challenging, since few enjoy having their work criticized and flaws identified, but is a necessary component of an overall secure software strategy. This role is critical to not only the success of the company, but also to all the customers who implement your software. Secure software development has a direct and undeniable impact on the ability of an organization to protect their systems and information assets, and you play a key role in that success.
    Details about recommended courses can be found at http://www.sans.org/20coolestcareers/#job18.

EC-Council

- **Why Secure Programming?**
  - According to the CERT/CC vulnerability reports, most of the vulnerabilities stem from a relatively small number of common programming errors.
  - Understanding the security vulnerabilities in the application allows designers to create better security strategies for the application. The pervasiveness of easily remedied vulnerabilities indicates a lack of developer education on secure coding - VERACODE. Security quality of software ultimately determines the rate of data breaches and cyber threats that result in substantial losses to an organization.

» Why Secure Programming is important?

- » According to the survey 'State of Web Application Security' conducted by Ponemon Institute and sponsored by Imperva & WhiteHat Security (published April 26, 2010), despite having mission-critical applications accessible via their websites, many organizations are failing to provide sufficient resources to secure and protect Web applications important to their operations. A study by IBM's System Sciences Institute found that the relative cost of fixing software defects after deployment is almost 15 times greater than detecting and eliminating them during development. Secure applications are also more protected from media criticism, more attractive to users, and less expensive to fix and support.

» How to be a successful Secure Programmer?

- » As in "criminal profiling", a process where descriptions and characteristics of unknown criminals are sketched, to be a successful secure programmer, you are expected to understand all the facets and the attack vectors a cyber – criminal may use to exploit the vulnerabilities in a software code. Most of the application developers do not have any formal training. Most of the programming education curriculums often do not include security issues, this result in developers who are either not aware of security issues or poorly skilled to overcome these issues. Educational and training centers have designed their teaching methodology such that it leaves a perception of security topics as a dull subject. The plan to enable the development and procurement of resilient software and systems throughout the lifespan of a project; where security and privacy requirements are defined at the beginning requires developers to understand software development lifecycle thoroughly and experience. Details about recommended courses can be found at http://www.eccouncil.org/certification.aspx.

(ISC)^2

The (ISC)² Resource Guide for Information Security Professional lists the latest resources in educational references, event listings, and leading industry organizations. The guide can be found at https://www.isc2.org/resourceguide/.

Sample Job Descriptions

» Cyber Software Assurance Developer/Integrator

- ○ Experience with applying security activities within SDLC
- ○ Experience with security, including CISSP or SANS secure programming assessments
- ○ Experience with security standards, including SSE-CMM, NIST SPs, ISO 15408 Common Criteria, or client-specific software assurance guides

» Software Assurance Engineer

- ○ Provide technical leadership in all aspects of software assurance and computer systems engineering support
- ○ Lead and actively participate in the evaluation and analyses of activities related to all phases of the secure software life cycle from initial planning, requirements definition, design and development through integrated system testing and sustaining operations.
- ○ Responsibilities will also include the support of a wide range of technical and programmatic activities for program offices, including leading the review and assessment of software system architecture; system requirements and their allocation to lower level specifications; design, code and test activities; trade studies; COTS/GOTS products; reuse software; test tools; simulators; software verification and validation (V&V); and system test and integration. Support independent review efforts in analyzing and assessing system software and related development and testing activities.

Academic Curricula Samples (https://buildsecurityin.us-cert.gov/swa/wetwgdocs.html)

- Carnegie Mellon University CS curriculum at http://www.csd.cs.cmu.edu/education/bscs/index.html#curriculum.
- George Washington University CS curriculum at http://www.cs.ucdavis.edu/courses/exp_course_desc/index.html
- Massachusetts Institute of Technology EECS Undergraduate Program at http://www.eecs.mit.edu/ug/index.html.
- Master's program in Secure Software Systems at James Madison University at http://www.cs.jmu.edu/sss/.
- Stanford University CS curriculum at http://cs.stanford.edu/Courses/.
- University of California at Davis CS curriculum at http://www.cs.ucdavis.edu/courses/exp_course_desc/index.html.

Commercial Training Examples

- Aspect Security, Inc., Application Security Education and Training at http://www.aspectsecurity.com/training.htm.
- EC-Council, Application and Information Security, and Computer Forensic Investigation Training at http://www.eccouncil.org/. EC-Council University Master of Security Science (MSS) at http://www.eccuni.us/Academics/MasterofSecurityScience.aspx
- Foundstone, Inc., Education at http://www.foundstone.com/us/education-overview.asp.
- KRvW Associates, LLC., Training Services at http://www.krvw.com/training/training.html.
- LogiGear, Inc., Web and Software Application Security Testing at http://www.logigear.com/training/course_catalog/course.asp?courseId=20.
- Microsoft Corp., Clinic 2806: Microsoft® Security Guidance Training for Developers (and other courses) at https://www.microsoftelearning.com/eLearning/courseDetail.aspx?courseId=26043.
- Netcraft, Inc., Web Application Security Course at http://audited.netcraft.com/web-application-course.
- Next Generation Security Software, Ltd., Security Training at http://www.ngssoftware.com/consulting/training/.
- SecuRisk Solutions at http://www.securisksolutions.com/index.php/education/training/
- Security Innovation, Inc., Application Security Education at http://www.securityinnovation.com/services/education/index.shtml.
- The SANS Institute, Inc. at https://www.sans.org/.

Conclusion

The goal of this pocket guide is to make this content “actionable” such that it leads to the development of educational and training materials

This pocket guide compiles software assurance education and training resources aimed to ensure adequate coverage of requisite knowledge areas and the corresponding roles in the workforce. In doing so it draws upon contributing disciplines such as software engineering (including its many subdisciplines), systems engineering, project management, etc., to identify and acquire competencies associated with secure software.

The Software Assurance Pocket Guide Series is developed in collaboration with the SwA Forum and Working Groups and provides summary material in a more consumable format. The series provides informative material for SwA initiatives that seek to reduce software vulnerabilities, minimize exploitation, and address ways to improve the routine development, acquisition and deployment of trustworthy software products. Together, these activities will enable more secure and reliable software that supports mission requirements across enterprises and the critical infrastructure.

For additional information or contribution to future material and/or enhancements of this pocket guide, please consider joining any of the SwA Working Groups and/or send comments to Software.Assurance@dhs.gov. SwA Forums are open to all participants and free of charge. Please visit https://buildsecurityin.us-cert.gov for further information.

No Warranty

This material is furnished on an “as-is” basis for information only. The authors, contributors, and participants of the SwA Forum and Working Groups, their employers, the U.S. Government, other participating organizations, all other entities associated with this information resource, and entities and products mentioned within this pocket guide make no warranties of any kind, either expressed or implied, as to any matter including, but not limited to, warranty of fitness for purpose, completeness or merchantability, exclusivity, or results obtained from use of the material. No warranty of any kind is made with respect to freedom from patent, trademark, or copyright infringement. Reference or use of any trademarks is not intended in any way to infringe on the rights of the trademark holder. No warranty is made that use of the information in this pocket guide will result in software that is secure. Examples are for illustrative purposes and are not intended to be used as is or without undergoing analysis.

Reprints

Any Software Assurance Pocket Guide may be reproduced and/or redistributed in its original configuration, within normal distribution channels (including but not limited to on-demand Internet downloads or in various archived/compressed formats).

Anyone making further distribution of these pocket guides via reprints may indicate on the pocket guide that their organization made the reprints of the document, but the pocket guide should not be otherwise altered. These resources have been developed for information purposes and should be available to all with interests in software security.

For more information, including recommendations for modification of SwA pocket guides, please contact Software.Assurance@dhs.gov or visit the Software Assurance Community Resources and Information Clearinghouse: https://buildsecurityin.us-cert.gov/swa to download this document either format (4"x8" or 8.5"x11").

Software Assurance (SwA) Pocket Guide Series

SwA is primarily focused on software security and mitigating risks attributable to software; better enabling resilience in operations. SwA Pocket Guides are provided; with some yet to be published. All are offered as informative resources; not comprehensive in coverage. All are intended as resources for 'getting started' with various aspects of software assurance. The planned coverage of topics in the SwA Pocket Guide Series is listed:

SwA in Acquisition & Outsourcing

I. Software Assurance in Acquisition and Contract Language
II. Software Supply Chain Risk Management & Due-Diligence

SwA in Development

I. Integrating Security into the Software Development Life Cycle
II. Key Practices for Mitigating the Most Egregious Exploitable Software Weaknesses
III. Risk-based Software Security Testing
IV. Requirements & Analysis for Secure Software
V. Architecture & Design Considerations for Secure Software
VI. Secure Coding & Software Construction
VII. Security Considerations for Technologies, Methodologies & Languages

SwA Life Cycle Support

I. SwA in Education, Training & Certification
II. Secure Software Distribution, Deployment, & Operations
III. Code Transparency & Software Labels
IV. Assurance Case Management
V. Assurance Process Improvement & Benchmarking
VI. Secure Software Environment & Assurance Ecosystem

SwA Measurement & Information Needs

I. Making Software Security Measurable
II. Practical Measurement Framework for SwA & InfoSec
III. SwA Business Case & Return on Investment

SwA Pocket Guides and related documents are freely available for download via the DHS NCSD Software Assurance Community Resources and Information Clearinghouse at https://buildsecurityin.us-cert.gov/swa.