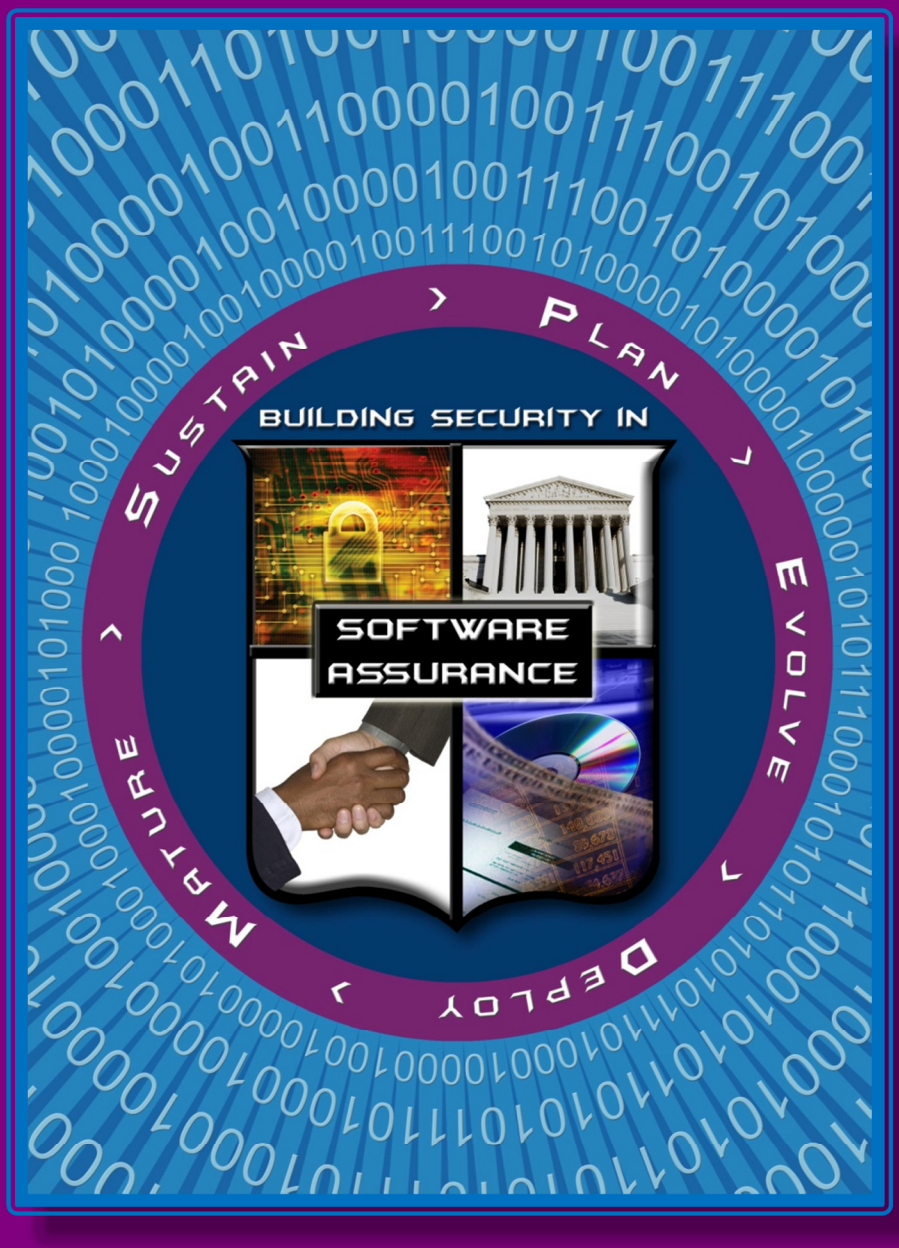
Software Assurance in Education, Training & Certification

Software Assurance Pocket Guide Series:
Life Cycle Support, Volume I
Version 2.1, March 1, 2011



Software Assurance (SwA) Pocket Guide Resources

This is a resource for 'getting started' in educating, training and certifying a workforce to build secure software. It describes how to promote awareness of the engineering activities and knowledge areas needed to build software that operates as expected, free from vulnerabilities. It summarizes how to train to prevent vulnerabilities from being intentionally designed into the software or accidentally inserted at any time during its life cycle. To do so, this guide describes knowledge areas for software assurance, starting with the core areas of study and extending to sub-disciplines to enhance with software security subject materials. It then presents lists of resources for accomplishing such study, including programs, tools, and books, with pointers on their use. Lastly, this guide describes the people who make up a security-conscious system development team, their education, titles, credentials, and standards. As part of the Software Assurance (SwA) Pocket Guide series, this resource is for information only. For details, see referenced source documents. For proper attribution, please include mention of these sources when referencing any part of this document

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.

Acknowledgements

The SwA community collaborates to develop SwA Pocket Guides. The SwA Forum and Working Groups function as a stakeholder meta-community that welcomes additional participation in advancing and refining software security. All SwA-related information resources are offered free for public use. The SwA community invites your input: please contact Software.Assurance@dhs.gov for comments and inquiries. For the most current pocket guides, refer to the SwA community website at https://buildsecurityin.us-cert.gov/swa/.

Members from government, industry, and academia comprise the SwA Forum and Working Groups. The Groups focus on incorporating SwA considerations into acquisition and development processes to manage potential risk exposure from software and from 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 knowledgeable workforce. One that is ready to perform engineering or technical activities that promote 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), July 2007 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,* September 1989 at http://satc.gsfc.nasa.gov/assure/agb.txt.
- "*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), September 2008 athttp://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, December 2005 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 athttp://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, Mead, N.R., Shoemaker, D., & Ingalsbe, J.A., January 2008 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, August 2010 athttp://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

Table of Contents

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 touches on software engineering (including its many sub-disciplines), systems engineering, project management, and other disciplines (shown in chart

Guiding Questions for SwA Curriculum Development:

- *Activities:* What engineering activities or aspects of activities relate to achieving secure software?
- *Knowledge:* What knowledge is needed to perform these activities or aspects?

Key SwA Knowledge Areas and Efforts, page 6). The goal is to fit the workforce with the ability to identify and acquire the 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 to identify strategies to inject software assurance related topics in the existing education and training programs.

The objective of software assurance is to ensure that the processes, procedures, and products used to produce and sustain the software conform to all specified requirements and standards. 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 that 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?

This Pocket Guide presents a general map of the areas of knowledge to cover in order to build security into software. The guide organizes the resources available for SwA outreach by avenue of approach: student curricula, workforce improvement, injection of subject-area material into related disciplines, credentialing, awareness, and independent study.

Resources

- Willis Ware, Security Controls for Computer Systems (U): Report of Defense Science Board Task Force on Computer Security; Rand Report R609-1, The RAND Corporation, Santa Monica, CA, Feb. 1970.
- James P. Anderson, Computer Security Technology Planning Study, ESD-TR-73-51, ESD/AFSC,

The Case for Software Assurance Education

Software assurance has become critical because dramatic increases in business and mission risks are now attributable to exploitable software.

- Software size and complexity obscures intent and precludes exhaustive testing.
- Outsourcing and use of un-vetted software supply chains increases risk.
- Attack sophistication now eases exploitation.

These factors 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 under attempted exploitation).

In their report to the President, 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."*

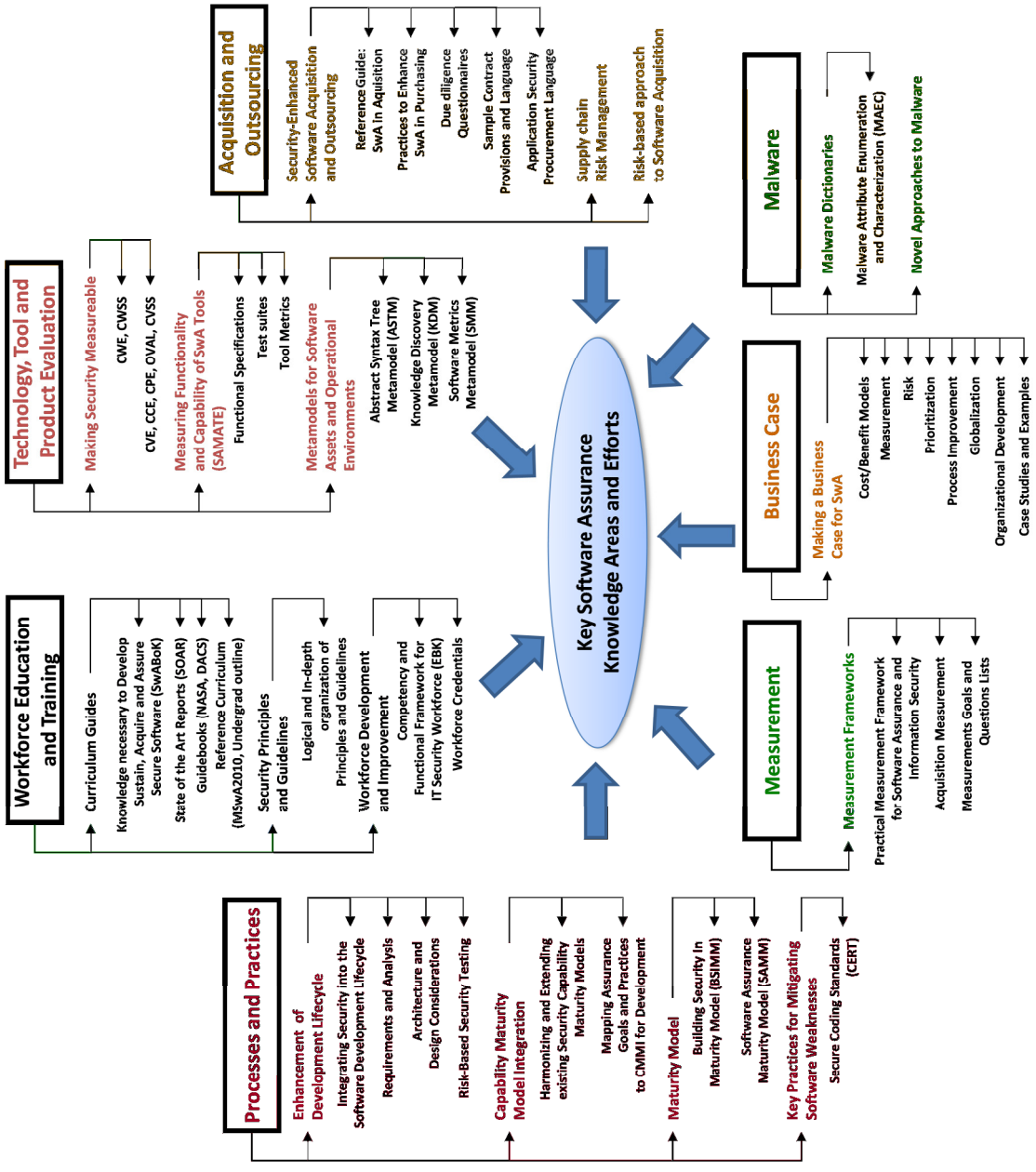
It is clear that to produce, acquire, and sustain secure software, a framework that identifies workforce needs for competencies, leverages sound practices, and guides curriculum development for education and training relevant to software assurance is inevitable. Because software quality assurance and 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.

Resources

- President's Information Technology Advisory Committee (PITAC) Report, Cyber Security: A Crisis of Prioritization, February 2005

Key SwA Knowledge Areas and Efforts

The chart below summarizes available workforce training materials and shows where to look for resources in SwA-related fields.



Curriculum and Training Guides

Table 1– SwA Curriculum and Training Development Guides		
Identifier	**Relevant Documents and Links**	**Purpose**
SwA Curriculum Project[1]	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 and Organization of Principles and Guidelines	Software Assurance Body of Knowledge. Version 1.2, Samuel T. Redwine, Jr. (Editor), DHS, https://buildsecurityin.us-cert.gov/bsi/dhs/927-BSI.html	Provides 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	Provides an extensive set of software system security principles and guidelines organized in a logical, in-depth fashion. Last updated February **2008**.

[1] The Institute of Electrical and Electronics Engineers (IEEE) Computer Society announced in December 2010 its recognition of the Master of Software Assurance (MSwA) Reference Curriculum. The Reference Curriculum identifies a core body of knowledge for MSwA degree program development. Sponsored by DHS NCSD, this MSwA project was led by educators from Carnegie Mellon University CERT in collaboration with educators from Embry-Riddle Aeronautical University, Monmouth University, and Stevens Institute of Technology. This project recognizes software assurance as an emerging and important multi-discipline body of knowledge: a field that covers how to provide the requisite levels of dependability and security throughout software development, acquisition, and operation . See materials at https://buildsecurityin.us-cert.gov/bsi/1165-BSI.html and www.cert.org/mswa. To facilitate implementation, the MSwA project team is available to assist educational institutions in starting an MSwA degree program, credential, or track based on the curriculum. This assistance, provided at no charge, includes review of implementation plans and mentoring to guide through implementation.

Workforce Development and Improvement

Table 2– Workforce Development and Improvement		
Identifier	**Relevant Documents and Links**	**Purpose**
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**.
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 of essential knowledge and skills that IT security practitioners should have in order to perform specific roles and responsibilities. Last updated September**2008.**
Information Security Workforce Development Matrix Project – Information Security Systems and Software Development Professional Role	See Section *Role Descriptions* section, page *17*, for materials from the matrix project developed by Federal CIO Council's IT Workforce Committee and Information Security and Identity Management Committee. Contacts: http://www.cio.gov/committees.cfm/csec/3/cid/4 http://www.cio.gov/committees.cfm/csec/3/cid/5	This project produces role-based information security workforce development matrices. The matrices are one-page dashboards intended to establish a baseline across the Federal Government for staff engaged in information security work. This initiative provides a government-wide perspective on common information security roles. The ISS&SDP is one of 11 roles that have been identified to date. Each matrix contains a uniform framework, by performance level, describing the recommended competencies/skills, education, experience, credentials, and training for a particular role. The matrices provide guidance for federal agencies and do not replace OPM basic qualifications.

Strategies for Injecting SwA Knowledge

Table 3– Strategies for Injecting SwA Knowledge Areas into existing Education and Training Programs	
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 curriculums are available at http://repository.cmu.edu/sei/3/ and http://repository.cmu.edu/sei/4/ Graduate Certificates and Master Degree Concentrations at the Stevens Institute of Technology:http://dc.stevens.edu/academic-programs/software-assurance/

Table 3– Strategies for Injecting SwA Knowledge Areas into existing Education and Training Programs	
Strategy	**Relevant Documents and Links**
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 Also: graduate-level Software Assurance courses that cover secure software engineering activities during the SDLC are 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 to inject topics such as Misuse Cases and Assurance Cases into existing software engineering or information security courses.
Capstone and Class Projects	Software Engineering capstone courses or class projects which 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 address three discipline- specific tracks. The targets are everyday non-technical computer users, technical IT professionals, 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/
	Podcasts http://www.cigital.com/silverbullet/

Table 3– Strategies for Injecting SwA Knowledge Areas into existing Education and Training Programs	
Strategy	**Relevant Documents and Links**
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 **Software Assurance Mobile Instructional (device) SAMI** Dan Shoemaker, University of Detroit Mercy A catalogue of available software assurance materials that are packaged and delivered using an iPad based instruction medium for educators
Credentialing	Several certification options are now available to suit the needs of specific job functions required in an enterprise. More information can be found in the Workforce Credentials section of this guide.

Tools

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

Table 4 – 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.	Misuse cases, security focused UML class diagrams and other documentation for class assignments and projects.
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 the combination of manual and automated techniques to enumerate, analyze, scan, attack and exploit web applications.
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 dont's.
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.
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.

Table 4 – Tools and web resources for hands-on classroom experience with SWA Concepts		
Tool Name	**Tool Description**	**Possible Classroom Uses**
Olly Debug	OllyDbg is a 32-bit assembly level debugger for Microsoft Windows. Located at www.ollydbg.de/	Emphasize binary code analysis and is particularly useful in cases where source is unavailable. Explain Buffer Overflows.
Pharos	Pharos 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	Pharos can be used as an introduction to web application security assessment.
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.
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.
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.
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.
Web Resources		
Google Code University	http://google-gruyere.appspot.com	Web application exploits and defenses. Topics include cross-site scripting, cross site request forgery, AJAX vulnerabilities, denial of service, etc.
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.
OWASP Broken Web Applications Project	http://www.owasp.org/index.php/OWASP_Broken_Web_Applications_Project	A collection of applications with known vulnerabilities.
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.

Books

Table 5 – 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 as a textbook or additional reference material.
	The Security Development Lifecycle: SDL: A Process for Developing Demonstrably More Secure Software, Michael Howard, Steve Lipner, Microsoft Press, 2006	Adaptation of Microsoft's Security Development Lifecycle (SDL) with case-studies on several Microsoft products.
	Secure and Resilient Software Development, Mark S. Merkow, Lakshmikanth Raghavan**,** Auerbach Publications, 2010	A practitioner's perspective on enterprise assurance programs.
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 as a textbook or additional reference material
	Secure Programming for Linux and Unix HOWTO, David Wheeler, 2003	Unix systems-specific guidelines for C, C++, Java, Perl, PHP, Python, Tcl, and Ada95. http://www.dwheeler.com/secure-programs/
	Secure Coding in C and C++, Robert Seacord, Addison-Wesley Professional, 2005	Examples of secure code, insecure code, and exploits, implemented for Windows and Linux. http://www.cert.org/books/secure-coding/
	24 Deadly Sins of Software Security: Programming Flaws and How to Fix Them Michael Howard, David LeBlanc, John Viega, McGraw-Hill Osborne Media, 2009	Identifies the most common design and coding errors, their fixes and avoidance strategies.
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/

Table 5 – A List of SwA focused Books for Use in Education and Training		
Topic	**Title and Publisher**	**Summary and Possible Use**
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 6– Domain-specific SwA standards used in practice		
Standard	**Source**	**Purpose**
Assurance Process Reference Model (PRM)	Presentation: https://buildsecurityin.us-cert.gov/swa/downloads/ACSAC2010BartolMoss12-05-2010.pdf Self Assessment: https://buildsecurityin.us-cert.gov/swa/downloads/20100922_PRM_Practice_List_2page.pdf	The Assurance PRM can be used to help organizations conduct a gap analysis of existing practices. The results of a gap analysis can be used to prioritize and track SwA implementation efforts. The Assurance PRM addresses assurance from executive to developer.
BSIMM2: The Building Security In Maturity Model	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.
CERT Resilience Management Model	http://www.cert.org/resilience/rmm.html	It has two primary objectives: 1. Establish the convergence of operational risk and resilience management activities such as security, business continuity, and aspects of IT operations management into a single model. 2. Apply a process improvement approach to operational resilience management through the definition and application of a capability level scale that expresses increasing levels of process improvement.

Table 6– Domain-specific SwA standards used in practice		
Standard	**Source**	**Purpose**
CMMI-ACQ: Carnegie Mellon University (CMU)/Software Engineering Institute (SEI) Capability Maturity Model Integration (CMMI) for Acquisitions	http://www.sei.cmu.edu/cmmi/index.cfm	Version 1.2. CMMI for Acquisition (CMMI-ACQ) is based on the CMMI Framework. CMMI for Acquisition (CMMI-ACQ) provides guidance to acquisition organizations for initiating and managing the acquisition of products and services. The model focuses on acquirer processes and integrates bodies of knowledge that are essential for successful acquisitions.
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++.
openSAMM: Open Web Application Security Project (OWASP) Open 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

Table 7– 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	Certified Secure Software Lifecycle Professional (**CSSLP**)	http://www.isc2.org/csslp-certification.aspx

Vendors

Educators and trainers are often faced with a question from students regarding possible roles in an enterprise for software assurance professionals and their demand in the workforce. To facilitate such discussion, we have collected a few quotes from several training and credentialing entities below.

SANS

The SANS Software Security Institute (SSI) brings computer security training to developers, programmers, and application security professionals. They offer training for web application security and hacking defense, secure coding, software security testing, code review, PCI compliance, and language specific training for Java/JEE, .NET, C, and others. They also offer the Programmer/Developer Certification (GSSP) through their GIAC affiliate. They are also a source of free resources such as the Top 25 Most Dangerous Programming Errors. According to the System Administration, Audit, Network Security (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, and identify 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 beyond the success of the company to that of 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.

EC-Council

The International Council of E-Commerce Consultants (EC-Council) is a member-based organization that certifies individuals in various e-business and information security skills. It is the owner and developer of the Certified Ethical Hacker (C|EH), Computer Hacking Forensics Investigator (C|HFI), and EC-Council Certified Security Analyst

(E|CSA)/License Penetration Tester (L|PT) programs, and various others offered in over 60 countries around the globe. EC-Council is the creator of the Hacker Halted conference and workshop series and has established the EC-Council University based in New Mexico, which offers both bachelor's and master's degree programs.

» Why Secure Programming

» **Most vulnerabilities stem from a relatively small number of common programming errors**, according to the CERT/CC vulnerability reports. 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.

» **Reduce costs:** The relative cost of fixing software defects after deployment is almost 15 times greater than detecting and eliminating them during development, according to a study by IBM's System Sciences Institute.

» **Reduce serious risks**: 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, according to the survey 'State of Web Application

» **Everyone benefits:** Secure applications are more protected from media criticism, more attractive to users, and less expensive to fix and support.

» How to Be a Successful Secure Programmer

» **Understand the adversary**: Just as a law officer uses "criminal profiling" to recognize the characteristics of unknown criminals, the successful secure programmer understands all the facets and the attack vectors a cyber-criminal may use to exploit the vulnerabilities in a software code.

» **Avoid training gaps**: Most application developers do not have any formal training in secure coding. Programming education curriculums most often do not include security issues. This results in developers who are either not aware of security issues or poorly skilled to overcome these issues.

» **Invigorate security topics**: The teaching methodology used by educational and training centers contributes to the erroneous perception of security topics as a dull subject. Incorporating the development and procurement of resilient software and systems throughout the lifespan of a project relies on developer interest

» **Work from a solid base**: Define security and privacy requirements at the beginning and require team members to understand the software development lifecycle thoroughly. Details about recommended courses can be found at http://www.eccouncil.org/certification.aspx.

(ISC)²

International Information Systems Security Certification Consortium, Inc., (ISC)²® (pronounced "ISC-squared") is a non-profit organization that educates and certifies information security professionals throughout their careers. The most widely known certification offered by the organization is the Certified Information Systems Security Professional (CISSP). The Certified Secure Software Lifecycle Professional (CSSLP) Certification Program shows software lifecycle stakeholders how to implement security and how to glean security requirements, design, architect, test, and deploy secure software. Attendees of the Software Assurance Forums and Working Group Sessions can receive "Continuing Professional Education" (CPE) credits.

Resources

- System Administration, Audit, Network Security (SANS). Details about recommended courses can be found at http://www.sans.org/20coolestcareers/#job18.
- *State of Web Application Security* conducted by Ponemon Institute and sponsored by Imperva & WhiteHat Security, published April 26, 2010.
- *The (ISC)² Resource Guide for Information Security Professional,* latest educational references, event listings, and leading industry organizations. https://www.isc2.org/resourceguide/

Role Descriptions

Cyber Software Assurance Developer/Integrator

- Experience with applying security activities within SDLC
- Experience with security, including CSSLP, CISSP and SANS secure programming assessments
- Experience with security standards, including SSE-CMM, NIST SPs, ISO 15408
- Common Criteria, or client-specific software assurance guides. (Also see the section on "Standards of Practice")

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.

Information Security Systems & Software Development Professional (ISSSDP)

- The Information Security Systems and Software Development Professional is responsible for secure design, development, testing, integration, implementation, sustainment, and/or documentation of software applications (web based and non-web) following formal secure systems development lifecycle processes and using security engineering principles.
- The following professional requirements form part of a broad Federal Government effort to identify and describe roles.
- **I: Entry level:**
  - Basic understanding of computer systems and related information security software and hardware components, network systems and databases, and information systems security safeguards

- Capable of articulating software/system abuse and misuse cases; understands practices and tools for mitigating exploitable software weaknesses
- Writes code in different programming languages (e.g., Java, C, C++)
- Integrates off-the-shelf products with information assurance safeguards (e.g., implementing network firewalls and routers)
- Participates in small teams performing software development and information security software-oriented tasks

» **II: Intermediate level:**

- Advanced understanding of information systems security, ethical hacking, multiple network analysis, configuration management, integration and deployment issues
- Designs secure software systems from requirements within the software development lifecycle; derives additional security requirements based on the deployment environment
- Communicates technical information to non-technical audiences and advises staff on information security issues and approaches
- Applies knowledge of information security aspects (e.g., coding, operating systems, programming languages, databases, federal and agency policies and procedures)
- Performs static and dynamic analysis to identify vulnerabilities in applications, across databases, networks, network-based environments, and operating systems, and directs remediation as appropriate
- Creates protocols, procedures, and guidelines to mitigate security risks

» **III: Advanced level:**

- Expert understanding of information systems security, ethical hacking, multiple network analysis, configuration management, integration and deployment issues
- Serves as senior advisor to the architectural design and development of enterprise-wide applications, systems, and services
- Serves as senior advisor to procurement and contract management in support of system and software acquisition
- Coordinates and collaborates across organizational and dept./agency lines; influences others to comply with security policies, standards, and best practices
- Integrates policies and procedures across government agencies; participates in the update of the agency's knowledge of federal security policies, procedures, and guidelines

» **Relevant Competency/Skills Sources**(as determined by the department/agency):

- OPM GS-2200 Job Family Standard Competencies
- Clinger-Cohen Core Competencies with an emphasis on Technical, Desktop Technology Tools, and IT Security/Information Assurance competency areas
- Essential
- (EBK) for IT Security Professionals (e.g., DHS and DOE)
- NIST Special Publications (e.g., NIST SP 800-16, Revision 1, 800-37, 800-53, 800-53A)
- ODNI Cyber Subdirectory Competencies
- OPM's Executive Core Qualifications (ECQs) (for SES positions)
- DHS SwA Communities and Forum (https://buildsecurityin.us-cert.gov/swa/)

» **References:** see Table 2, ***Information Security Workforce Development Matrix Project*** ,above, Page *8*, "

Table 8– Credentials, Competencies and Skills by Performance Level			
	Entry Level	**Intermediate Level**	**Advanced Level**
Software Development **Written & Oral Communication** **Creative Problem Solving** **Information Security/Assurance** **Critical Thinking and Analytical Skills**	Yes	Yes	Yes
Software Engineering **Project/Program Management** **Leadership & People Management**	No	Yes	Yes
Suggested Credentials in: ▪ **Computer science/engineering** ▪ **Database/information management** ▪ **Information assurance/security** ▪ **Software assurance/security** ▪ **Information systems management**	Associate's Degree from an accredited program	Bachelor's Degree from an accredited program	Master's Degree from an accredited program plus 5 years' experience
OR alternative experience/credentials: ▪ **Information systems security certifications (e.g., ISC² CSSLP and CISSP, various SANS certs, EnCase Cybersecurity suite)** ▪ **Network engineering certifications (e.g., EnCase forensics suite, Cisco Certified Internetwork Expert (CCIE), GIAC Security Essentials Certification GSEC)** ▪ **Software/systems administration/engineering/development/ Database Management System (DBMS)certifications** ▪ **Relevant programming languages experience**	1 year experience in work directly related to: ▪ Secure network design, ▪ Database design and security, ▪ Secure coding and testing techniques	6 years experience Possession and demonstrated application of relevant certifications (as determined by the department/agency)	Bachelor's degree Plus 10 years relevant work experience and project management experience Possession and demonstrated application of relevant certifications (as determined by the department/agency), with the addition of PMP

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 athttp://www.cs.jmu.edu/sss/.
- Stevens Institute of Technology, Software Assurance Program at http://sse.stevens.edu/academics/graduate/software-engineering/program-overview/software-assurance/
- 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 promote the development of educational and training materials and programs to prepare a workforce more capable of securing software and applications.

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 sub-disciplines, such as programming), systems engineering and analysis, 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, 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 back cover of 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
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.

SwA Community Resources & Information Clearinghouse
https://buildsecurityin.us-cert.gov/swa