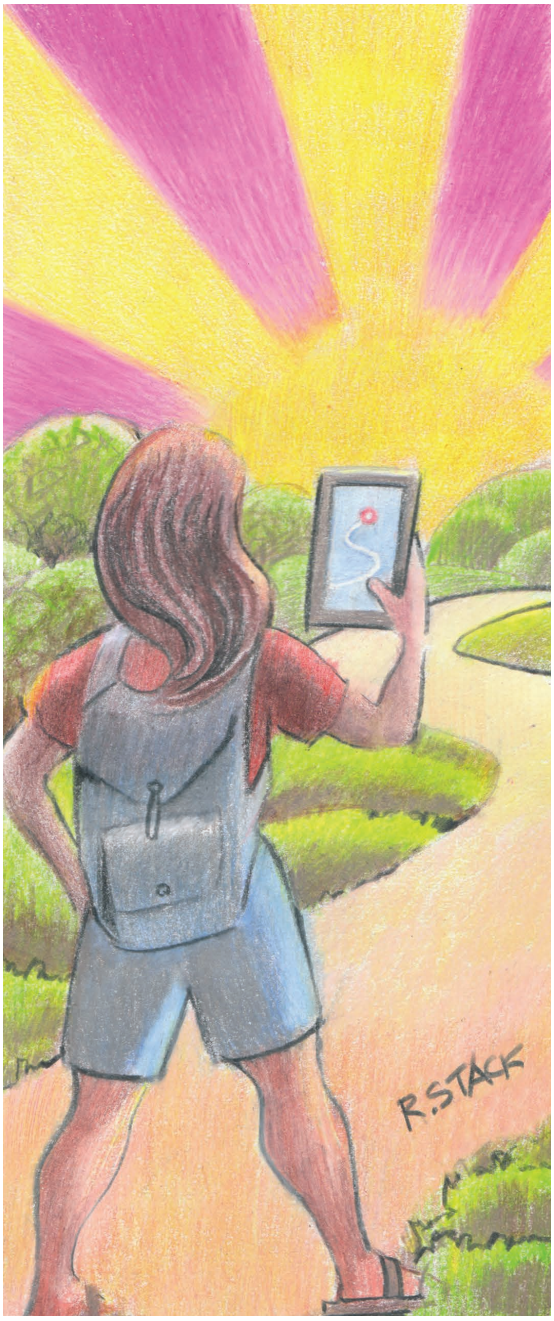
Editors: Jonathan Margulies, margulies@gmail.com | Luanne Goldrich, luanne.goldrich@jhuapl.edu

Building Security In:
Preparing for a Software Security Career

Nancy R. Mead | Carnegie Mellon University Software Engineering Institute
Thomas B. Hilburn | Embry-Riddle Aeronautical University



Because of software security problems and their impact on the proper functioning of major software systems, the interest in and demand for software security specialists has grown dramatically. The US Department of Homeland Security, the US Department of Defense, Carnegie Mellon University's Software Engineering Institute (SEI), and other government, commercial, and educational organizations are all interested in advancing the software security workforce's educational preparation. However, until recently, there has been little guidance on how to prepare for a software security career. Most specialists start with some sort of computing degree, maybe with a programming course that included topics in secure coding. Then, through on-the-job training and experience, they gain proficiency in certain aspects of building secure software. It can be a difficult and meandering road. Unfortunately, this approach doesn't provide the preparation needed to address current security risks.

In the previous installment of Building Security In,[1] we described a Software Assurance (SwA) Competency Model.[2] SwA is the

> application of technologies and processes to achieve a required level of confidence that software systems and services function in the intended manner, are free from accidental or intentional vulnerabilities, provide security capabilities appropriate to the threat environment, and recover from intrusions and failures.[3]

Two of the model's objectives are as follows:

- Enhance SwA curriculum guidance by providing information about industry needs and expectations for competent SwA professionals.
- Provide direction and a progression for SwA professionals' development and career planning.

Here, we discuss how to meet these objectives.

The Software Assurance Curriculum Project

The SEI established the Software Assurance Curriculum Project in 2009. The project has developed four documents that correlate well with the objective to enhance SwA curriculum guidance (see Table 1).

The courses listed in Table 1 go well beyond secure coding and SwA at the implementation level. They cover security issues throughout the life cycle, as part of requirements analysis, architecture and module design, implementation, testing, and operation and maintenance. The graduate level includes additional SwA topics in such traditional areas as management and process, requirements engineering,

 Copublished by the IEEE Computer and Reliability Societies

Table 1. Software Assurance Curriculum Project documents.

Document	Description
Volume I: Master of Software Assurance Reference Curriculum[3]	Provides material for establishing or revising a Master of Software Assurance (MSwA) program: curriculum development guidelines, graduate-student outcomes, recommended student preparation, an SwA body of knowledge, a high-level MSwA curriculum architecture, and implementation guidelines.
Volume II: Undergraduate Course Outlines[4]	Provides the syllabi for seven undergraduate SwA courses: Computer Science I and II, Introduction to Computer Security, Software Security Engineering, Software Quality Assurance, Software Assurance Analytics, and Software Assurance Capstone Project. Each syllabus contains a course description, prerequisite knowledge, a list of learning objectives and topics, sources for the course, course delivery features, and course assessment features.
Volume III: Master of Software Assurance Course Syllabi[5]	Provides the syllabi for nine graduate SwA courses: Assurance Management; System Operational Assurance; Assured Software Analytics; Assured Software Development 1, 2, and 3; Assurance Assessment; System Security Assurance; and Software Assurance Capstone Experience. The syllabi are organized similarly to those in volume II but include a schedule of weekly in-class activities, suggested readings, and out-of-class assignments.
Volume IV: Community College Education[6]	Provides the syllabi for six SwA courses appropriate for community college students: Computer Science I, II, and III; Introduction to Computer Security; Secure Coding; and Introduction to Assured Software Engineering.

Table 2. SwA CorBoK Knowledge Areas.

Knowledge area	MSwA student outcomes
Assurance across Lifecycles	The ability to incorporate assurance technologies and methods into life-cycle processes and development models for new or evolutionary system development and for system or service acquisition.
Risk Management	The ability to perform risk analysis and trade-off assessment and to prioritize security measures.
Assurance Assessment	The ability to analyze and validate the effectiveness of assurance operations and create auditable evidence of security measures.
Assurance Management	The ability to make a business case for software assurance, lead assurance efforts, understand standards, comply with regulations, plan for business continuity, and keep current in security technologies.
System Security Assurance	The ability to incorporate effective security technologies and methods into new and existing systems.
System Functionality Assurance	The ability to verify new and existing software system functionality for conformance to requirements and to help reveal malicious content.
System Operational Assurance	The ability to monitor and assess system operational security and respond to new threats.

design, construction, testing, and sustainment. These areas include SwA topics such as security policy and security functionality requirements; attack methods to damage software; analysis of threats to software; appropriate countermeasures such as layers, access controls, privileges, intrusion detection, and encryption; and designing and planning for access control, privileges, and authentication.

Because no SwA body of knowledge existed, one of the project team's first tasks was to establish one. After extensively reviewing software security reports, books, and articles and after surveys of and discussions with industry and government SwA professionals, the project team developed the SwA Core Body of Knowledge (CorBoK).[3] The CorBoK covers the spectrum of SwA practices involved in software system acquisition, development, operation, and evolution. It's the source for the content of the courses listed for volumes II, II, and IV in Table 1. Table 2 lists the CorBoK's principal components, *knowledge areas* (KAs), and describes the principal Master of Software Assurance (MSwA) student outcomes associated with each KA.

On the basis of the KAs, the project team created the MSwA Curriculum Architecture (see Figure 1). This architecture is compatible with software engineering master's programs because software engineering courses can incorporate the SwA-specific topics. The MSwA core and capstone experience in Figure 1 constitute the courses in volume III; in total, they cover all the KAs in

Preparatory materials	Computing foundations Software engineering Security engineering
MSwA core	Assurance Management System Operational Assurance Assured Software Analytics Assured Software Development 1, 2, and 3 Assurance Assessment System Security Assurance
Electives	Courses related to assurance in selected domains
Capstone experience	Project

Figure 1. The Master of Software Assurance (MSwA) Curriculum Architecture. It provides a structural basis for programs that deliver the outcomes described in Table 2.

Table 2. The architecture provides a structural basis for programs that deliver the outcomes described in Table 2. Of course, programs might cover the SwA body of knowledge and the corresponding outcomes using a different organization and set of courses than are in Figure 1. Figure 1 also lists the three preparatory areas students need to pursue the MSwA: computing foundations, software engineering, and security engineering. Volume I describes these areas in detail.

Using the SwA Curricula

Although some programs are using our course and curriculum guidance, the SwA curricula aren't yet widely implemented. We're still in the outreach-and-dissemination phase, one of the purposes of this article. However, people interested in perfecting their SwA competency could study the four volumes to plan their education or self-study. For instance, if you already have an undergraduate computing degree and are a software security specialist, studying volumes I and III will let you assess your skills and knowledge and plan for future study. Consider the following case study.

Joan graduated with a computer science BS a few years ago and is a software engineer for Fly-by-Night Airlines. As part of a team directed by the system architect, she develops and maintains small and mid-sized units for a software system that provides services for passengers and flight crews.

In her undergraduate education, Joan acquired most of the SwA skills and knowledge described in the Computer Science I and II courses, which are part of volume II—for example, foundations of information security, design concepts and principles, design by contract, exception handling, secure programming, coding standards, algorithm and code review, unit test design, penetration testing, program metrics, and quality assessment. In her current position, Joan has practiced these skills and participated in employer-sponsored workshops and training sessions.

However, Joan would like to advance and acquire additional SwA knowledge and skills. She has looked through the SwA Competency Model and identified where she needs further professional development. So, she reads through volumes I and III to see what to study. She notices that the Assured Software Development 1 course covers several topics that she is weak in and would like to learn more about: software processes, requirements engineering, software architecture, and software security topics such as assurance risk assessment, attack trees, and misuse or abuse cases.

On the basis of her analysis of volumes I and III, Joan examines a local university's courses, looking for ones that will help her in the areas she would like to study. She finds courses that cover software process, requirements engineering, and software architecture. However, she can't find anything that includes the other topics on her wish list.

Joan looks back at Assured Software Development 1 and reviews the description of the primary sources recommended for it. She purchases both books listed and uses them as part of her study plan. She consults with her supervisor about taking courses at the local university and pursuing self-study using those books. Her supervisor makes a few minor suggestions and strongly encourages her to proceed with her plan.

As Joan proceeds through her self-study, she improves her software security knowledge and capability and can apply this in her work. Her supervisor notices and comments on Joan's improved SwA competency.

We believe that Joan's situation, although fictitious, is realistic. We hope it would translate to many other situations in which individuals seek to assess or advance their SwA knowledge and skills.

The SEI developed the SwA curricula and course materials to create a foundation for establishing SwA programs and enhancing other computing programs with SwA topics and courses. We believe these materials can promote changes in computing curricula that will

significantly improve the preparation of software security specialists. These changes, especially in conjunction with the competency model, will enhance the staffing and professional development of software security specialists. This in turn will lead to a more robust, professional approach to preventing and solving software security problems. ■

Acknowledgments

We appreciate the support of the Software Assurance Curriculum team and Joe Jarzombek at the US Department of Homeland Security. This material is based on research funded and supported by the US Department of Defense under contract FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. This material has been approved for public release and unlimited distribution. DM-0000587.

References

1. T.B. Hilburn and N.R. Mead, "Building Security In: A Road to Competency," *IEEE Security & Privacy*, vol. 11, no. 5, 2013, pp. 89–92.
2. T. Hilburn et al., "Software Assurance Competency Model," tech. note CMU/SEI-2013-TN-004, Software Eng. Inst., Carnegie Mellon Univ., Mar. 2013; www.sei.cmu.edu/library/abstracts/reports/13tn004.cfm.
3. N.R. Mead et al., *Software Assurance Curriculum Project Volume I: Master of Software Assurance Reference Curriculum*, tech. report CMU/SEI-2010-TR-005, Software Eng. Inst., Carnegie Mellon Univ., Aug. 2010; www.sei.cmu.edu/library/abstracts/reports/10tr005.cfm.
4. N.R. Mead et al., *Software Assurance Curriculum Project Volume II: Undergraduate Course Outlines*, tech. report CMU/SEI-2010-TR-019 or ESC-TR-2010-019, Software Eng. Inst., Carnegie Mellon Univ., Aug. 2010; www.sei.cmu.edu/library/abstracts/reports/10tr019.cfm.
5. N.R. Mead et al., *Software Assurance Curriculum Project Volume III: Master of Software Assurance Course Syllabi*, tech. report CMU/SEI-2011-TR-013 or ESC-TR-2011-013, Software Eng. Inst., Carnegie Mellon Univ., Mar. 2011; www.sei.cmu.edu/library/abstracts/reports/11tr013.cfm.
6. N.R. Mead et al., *Software Assurance Curriculum Project Volume IV: Community College Education*, tech. report CMU/SEI-2011-TR-017 or ESC-TR-2011-017, Software Eng. Inst., Carnegie Mellon Univ., Sept. 2011; www.sei.cmu.edu/library/abstracts/reports/11tr017.cfm.

Nancy R. Mead is a principal researcher in the CERT Program at the Carnegie Mellon University Software Engineering Institute and an adjunct professor of software engineering at CMU. Contact her at nrm@sei.cmu.edu.

Thomas B. Hilburn is a professor emeritus of software engineering at Embry-Riddle Aeronautical University. Contact him at hilburn@erau.edu.