The Software Assurance Competency Model: A Roadmap to Enhance Individual Professional Capability

Nancy R. Mead
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania
United States
nrm@sei.cmu.edu

Dan Shoemaker
University of Detroit Mercy
Detroit, Michigan
United States
Dan.Shoemaker@att.net

Abstract

This paper describes a software assurance competency model that can be used by individual professionals to improve their software assurance skills. It can also be used by universities to align course content with skills needed in industry, and it can be used by industry to help employee professional growth as well as to screen prospective employees. The knowledge and skill areas in the competency model are based on the Master of Software Assurance reference curriculum that has been previously approved by the IEEE Computer Society and ACM. The model is aligned with a similar effort by the IEEE Professional Activities Board to develop a competency model for software engineering practitioners.

1. Competency and the Profession

Software was with us long before the creation of FORTRAN [1]. The roots of software engineering as a profession go back to the late 1960s and early 1970s, with the emergence of structured programming, structured design, and process models such as the Waterfall model [19]. What that means is that, at a minimum, software engineering has been a regular profession for at least 42 years.

In those four decades there have been numerous general attempts to define what a competent software professional should look like. Examples of this range from Humphrey's first published work on capability [12], through the effort to define software engineering as a profession, accompanied by a Software Engineering Body of Knowledge [13] and the People Capability Maturity Model [3].

The success of these efforts is still debatable, but one thing is certain: up to this point, there have been only a few narrowly focused attempts to define the professional qualities needed to develop a secure software product. The Software Assurance (SwA) Competency Model was developed to address this missing element of the profession.

The obvious question, given all of this prior work, is, "Why do we need one more professional competency model?" The answer lies in the significant difference between the competencies required to produce working code and those that are needed to produce software free from exploitable weaknesses. That difference is underscored by the presence of the adversary.

In the 1990s it was generally acceptable for software to have flaws as long as those flaws did not impact program efficiency or the ability to satisfy user requirements. So development and assurance techniques focused on proper execution with no requirements errors. Now, bad actors can exploit an unintentional defect in a program to cause all kinds of trouble. So although they are related in some ways, the professional competencies that are associated with the assurance of secure software merit their own specific framework.

A specific model for software assurance competency provides two advantages for the profession as a whole. First and most important, a standard model allows prospective employers to define the fundamental capabilities needed by their workforce. At the same time it will allow organizations to establish a general, minimum set of competency requirements for its employees; and more importantly it will allow companies to tailor an exact set of competency requirements for any given project.

From the standpoint of the individual worker, a competency model will provide software assurance professionals with a standard roadmap that they can use to improve performance by adding specific skills needed to obtain a position and climb the competency ladder for their profession. For example, a new graduate starting in an entry-level position could map out a path for enhancing their skills and planning their career advances as a software assurance professional. In many respects this latter feature makes a professional competency model a significant player in the development of the workforce of the future, which of course is of interest to software engineering educators and trainers.

2. The Software Assurance Competency Model

For the purposes of this Model, the following definition of *software assurance* will be used [14]:

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.

In the process of developing the Software Assurance Competency Model, a number of other competency models and supporting material were studied and analyzed [4, 2]. The Professional Advisory Board (PAB) of the IEEE Computer Society contributed a draft Framework for IEEE PAB Competency Models [20, 21]. This Framework offers an introduction to competency models and presents guidelines for achieving consistency among them. This is built around a generic structure for a professional area. It is then instantiated with specific knowledge, skills, and effectiveness levels for a particular computing profession, for instance, software engineering practitioner.

Other work on competency models consulted for the software assurance competency model include [7], [10], [18], [22], and [17]. “Balancing Software Engineering Education and Industrial Needs” [17] is an article that reports on a study that was conducted to help both academia and the software industry form a picture of the relationship between the competencies of recent graduates of undergraduate and graduate software engineering programs and the competencies needed to perform as a software engineering professional.

A key reference for the SwA Competency Model is the Master of Software Assurance Reference Curriculum [14]. The curriculum underwent both internal and public review, and was endorsed by both ACM and IEEE Computer Society as being appropriate for a Master’s degree

in Software Assurance. The curriculum document includes a mapping of the software assurance topic areas to GSwE2009 [7], thus providing a comparison to software engineering knowledge areas. Since then, elements of the curriculum have been adopted by various universities, including the Air Force Academy [8, 9], Carnegie Mellon University, Stevens Institute of Technology, and notably by $(ISC)^2$, a training and certification organization. As noted below, the MSwA Curriculum was the primary source for the knowledge and skills used in the Competency Model for various levels of professional competency.

The Software Assurance Competency Model will provide employers of software assurance personnel with a means to assess the software assurance capabilities of current and potential employees. In addition, along with the MSwA reference curriculum, this model is intended to guide academic or training organizations in the development of education and training courses to support the needs of organizations that are hiring and developing software assurance professionals.

The SwA Competency Model will enhance the guidance of software engineering curricula by providing information about industry needs and expectations for competent security professionals [14, 15, 16]; the Model will also provide software assurance professionals with direction and a progression for development and career planning. Finally, a standard competency model will provide support for professional certification activities.

2.1 SwA Competency Model Features

In the software assurance competency model, five levels (L1-L5) of competency are employed to distinguish different levels of professional capability, relative to knowledge, skills, and effectiveness [20]:

L1 - Technician

- Possesses technical knowledge and skills, typically gained through a certificate or an associate degree program, or equivalent knowledge and experience.
- May be employed in system operator, implementer, tester, and maintenance positions with specific individual tasks assigned by someone at a higher hierarchy level.
- Main areas of competency are SOA, SFA, and SSA. (see Table 1)
- Major tasks: low-level implementation, testing, and maintenance.

L2 - Professional Entry Level

- Possesses “application-based” knowledge and skills and entry-level professional effectiveness, typically gained through a bachelor’s degree in computing or through equivalent professional experience.
- May perform all tasks of L1 and additionally: manage a small internal project, supervise and assign sub-tasks for L1 personnel, supervise and assess system operations, and implement commonly accepted assurance practices.
- Main areas of competency are SFA, SSA, and AA. (see Table 1)
- Major tasks: requirements fundamentals, module design, implementation.

L3 - Practitioner

- Possesses breadth and depth of knowledge, skills, and effectiveness beyond the L2 level, and typically has two to five years of professional experience.
- May perform all tasks of L2 personnel and additionally set plans, tasks and schedules for in-house projects, define and manage such projects and supervise teams on the enterprise level, report to management, assess the assurance quality of a system, implement and promote commonly accepted software assurance practices.
- Main areas of competency are RM, AA, and AM. (see Table 1)
- Major tasks: requirements analysis, architectural design, tradeoff analysis, risk assessment.

L4 - Senior Practitioner

- Possesses breadth and depth of knowledge, skills, and effectiveness and a variety of work experiences beyond L3, with five to ten years of professional experience and advanced professional development, at the master's level or with equivalent education/training.
- May perform all tasks of L3 personnel and identify and explore effective software assurance practices for implementation, manage large projects, interact with external agencies, etc.
- Main areas of competency are RM, AA, AM, and AALC. (see Table 1)
- Major tasks: assurance assessment, assurance management, risk management across the life cycle

L5 - Expert

- Possesses competency beyond L4; advances the field by developing, modifying, and creating methods, practices, and principles at the organizational level or higher; has peer/industry recognition.
- Typically includes a low percentage of an organization's work force within the SwA profession (e.g., 2 % or less).

2.2 SwA Knowledge, Skills, and Effectiveness

The primary source for SwA Competency Model knowledge and skills is the Core Body of Knowledge (CorBoK), contained in *Software Assurance Curriculum Project, Volume I: Master of Software Assurance Reference Curriculum* [14]. The CorBoK consists of the knowledge areas listed in Table 1. Each knowledge area is further divided into second-level units as shown in Table 3. For each unit, competency activities are described for each of the levels L1-L5.

Table 1. CorBoK Knowledge Areas and Competencies

Knowledge Area (KA)	KA Competency
AALC: Assurance Across Life Cycles L3, L4, L5	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.
RM: Risk Management L2, L3, L4, L5	The ability to perform risk analysis and tradeoff assessment, and to prioritize security measures.
AA: Assurance Assessment L1, L2, L3, L4	The ability to analyze and validate the effectiveness of assurance operations and create auditable evidence of security measures.
AM: Assurance Management L3, L4, L5	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.
SSA: System Security Assurance L1, L2, L3, L4	The ability to incorporate effective security technologies and methods into new and existing systems.
SFA: System Functionality Assurance L1, L2, L3	The ability to verify new and existing software system functionality for conformance to requirements and to help reveal malicious content.
SOA: System Operational Assurance L1, L2, L3	The ability to monitor and assess system operational security and respond to new threats.

Other than a unit on "Ethics and Integrity" in the System Security Assurance Knowledge Area, the CorBoK does not contain topics on competency associated with effectiveness; the effectiveness attributes are listed in Table 2 (adapted from [20]). In Table 2, for a given attribute, there is no differentiation in effectiveness for the different competency levels; however, professionals would be expected to show an increase in the breadth and depth of capability in these areas of effectiveness as they proceed through their careers and move to higher competency levels.

Table 2. Competency Attributes of Effectiveness

Aptitude is exhibited by the ability do a certain software assurance activity at a certain level of competence. Aptitude is not the same as knowledge or skill but rather indicates the ability to apply knowledge in a skillful way. L2-L5
Initiative is exhibited by the ability to start and follow through on a software assurance work activity with enthusiasm and determination. L1-L5
Enthusiasm is exhibited by being interested in and excited about performing a software assurance work activity. L1-L5
Willingness is exhibited by undertaking a work activity, when asked, even if it is an activity the individual is not enthusiastic about performing. L1-L5
Communication is exhibited by expressing thoughts and ideas in both oral and written forms in a clear and concise manner while interacting with team members, managers, project stakeholders, and others. L2-L5
Teamwork is exhibited by working enthusiastically and willingly with other team members while collaborating on work activities. L1-L5
Leadership is exhibited by effectively communicating a vision, strategy, or technique that is accepted and shared by team members, managers, project stakeholders, and others. L3-L5

2.3 Competency Designations

Table 3 presents a portion of the CorBoK knowledge areas and second-level units, along with a description of the appropriate knowledge and skills for each competency level and the effectiveness attributes. The complete table can be found in the competency report [11]. A designation of L1 applies to levels L1 through L5; a designation of L2 applies to L2 through L5; and so on. The level descriptions indicate the competency activities that are demonstrated at each level.

Table 3. SwA Competency Designations

Knowledge/Skill/Effectiveness		
KA	**Unit**	**Competency Activities**
Assurance Across Life Cycles	Software Life-Cycle Processes	L1: Understand and execute the portions of a defined process applicable to their assigned tasks. L2: Manage the application of a defined life-cycle software process for a small internal project. L3: Lead and assess process application for small and medium sized projects, over a variety of life-cycle phases, such as new development, acquisition, operation, and evolution. L4: Manage the application of a defined life-cycle software process for a large project, including selecting and adapting existing SwA practices by life-cycle phase. L5: Analyze, design, and evolve life-cycle processes that meet the special organizational or domain needs and constraints.
	Software Assurance Processes and Practices	L1: Possess general awareness of methods, procedures, and tools used to assess assurance processes and practices. L2: Apply methods, procedures, and tools to assess assurance processes and practices. L3: Manage integration of assurance practices into typical life-cycle phases. L4: Lead the selection and integration of life-cycle assurance processes and practices in all projects, across an organization. L5: Analyze assurance assessment results to determine best practices for various life-cycle phases.
Risk Management	Risk Management Concepts	L1: Understand the basic elements of risk analysis. L2: Explain how risk analysis is performed. L3: Determine the models, process, and metrics to be used in risk management for small internal projects. L4: Develop the models, processes, and metrics to be used in risk management of any sized project. L5: Analyze the effectiveness of the use and application of risk management concepts across an organization.
	Risk Management Process	L1: Describe an organization risk management process. L2: Identify and classify the risks associated with a project. L3: Analyze the likelihood, impact, and severity of each identified risk for a project. Plan and monitor risk management for small to medium sized projects. L4: Plan and monitor risk management for a large project. L5: Develop a program for analyzing and enhancing risk management practices across an organization.
	Software Assurance Risk Management	L1: Describe risk analysis techniques for vulnerability and threat risks. L2: Apply risk analysis techniques to vulnerability and threat risks. L3: Analyze and plan for mitigation of software assurance risks for small systems. L4: Analyze and plan for mitigation of software assurance risks for both new and existing systems. L5: Assess software assurance processes and practices across an organization and propose improvements.

2.4 Examples of the Software Assurance Competency Model in Practice

There are a number of ways the Software Assurance Competency Model can be applied in practice. For example, an organization intending to hire an entry-level software assurance professional could examine the L1-L2 levels and incorporate elements of them into their job descriptions. These levels could also be used during the interview process by both the employer and the prospective employee, to assess the actual expertise of the candidate.

Another application is by faculty members who are developing courses in software assurance or adding software assurance elements to their software engineering courses. The use of the levels allows faculty to easily see the depth of content that is suitable for courses at the community college, undergraduate, and graduate levels. For example, undergraduate student outcomes might be linked to the L1 and L2 levels, whereas graduate courses aimed at practitioners with more experience might target higher levels. In industry, the model could be used to determine if specific competency areas were being overlooked. These areas could point towards corresponding training needs. With a bit of effort, trainers can tailor their course offerings to the target audience. The Model eliminates some of the guesswork involved in deciding what level of material is appropriate for a given course.

It can also be used by faculty who are already teaching such courses to assess whether the course material is a good fit for the target audience. The authors of this paper are currently teaching software assurance courses and can use the model to revisit and tailor their syllabi accordingly.

3. Summary and Future Plans

This Software Assurance Competency Model was developed to create a foundation for assessing and advancing the capability of software assurance professionals. The span of competency levels L1 through L5 and the decomposition into individual competencies based on the knowledge and skills described in the SwA CorBoK [14] provide the detail necessary for an organization or individual to determine SwA competency across the range of knowledge areas and units. The Model also provides a framework for an organization to adapt its features to the organization's particular domain, culture, or structure.

The Model has been reviewed by invited industry reviewers and mapped to actual industry positions. It has additionally undergone public review prior to publication. Our plan is to work closely with the IEEE Professional Advisory Board in order to be consistent with other similar documents, and to map the competency model to other selected competency models.

The most important outcome of this model will be a better trained and educated workforce. As the needs of the software industry for more secure applications continue, the recommendations of this model can be utilized to ensure better and more trustworthy practice in the process of developing and sustaining an organization's software assets. That guidance going forward is a critical linchpin in the overall effort to create trusted systems and provides the necessary reference to allow organizations and individuals to help achieve cybersecurity.

Acknowledgment

The authors appreciate the review and comments provided by Tom Hilburn, the lead author of the competency model, and the contributions of the other authors, Mark Ardis, Glenn Johnson, and Andrew Kornecki.

This material is based upon work funded and supported by the Department of Defense under Contract No. 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-0000160

References

[1] Backus, J. W.; H. Stern, I. Ziller, R. A. Hughes, R. Nutt, R. J. Beeber, S. Best, R. Goldberg, L. M. Haibt, H. L. Herrick, R. A. Nelson, D. Sayre, P. B. Sheridan, "The FORTRAN Automatic Coding System". Western joint computer conference: Techniques for reliability, Los Angeles, California: Institute of Radio Engineers, American Institute of Electrical Engineers, ACM: 188–198

[2] Behrens, Sandra; Alberts, Christopher; & Ruefle, Robin. *Competency Lifecycle Roadmap: Toward Performance Readiness* (CMU/SEI-2012-TN-020). Software Engineering Institute, Carnegie Mellon University, 2012. http://www.sei.cmu.edu/library/abstracts/reports/12tn020.cfm

[3] Curtis, B., Hefley, W.E., and Miller, S. (2002). *The People Capability Maturity Model: Guidelines for Improving the Workforce*. Reading, MA: Addison Wesley Longman

[4] *Software Assurance Professional Competency Model*, Department of Homeland Security, October 2012.

[5] *Information Assurance Workforce Improvement Program*, DoD Directive 8570.1, Department of Defense, December 19, 2005.

[6] *Information Technology Competency Model*, Employment and Training Administration, Department of Labor, August 2012.

[7] Graduate Reference Curriculum for Systems Engineering (GRCSE), version 0.5*, Body of Knowledge and Curriculum to Advance Systems Engineering (BKCASE)*, Hoboken, NJ, USA: Stevens Institute of Technology, December 2011. (http://www.bkcase.org/)

[8] Hadfield, S.; Schweitzer, D.; Gibson, D.; Fagin, B.; Carlisle, M.; Boleng, J.; & Bibighaus, D. "Defining, Integrating, and Assessing a Purposeful Progression of Cross-Curricular Initiatives into a Computer Science Program." Proceedings of the 41st ASEE/IEEE Frontiers in Education Conference. October 2011.

[9] Hadfield, S. "Integrating Software Assurance and Secure Programming Concepts and Mindsets into an Undergraduate Computer Science Program." Presented at *Department of Homeland Security Software Assurance Forum*. March 29, 2012.

[10] Hilburn, T. et al, *Software Engineering Competency Study: Final Report, Federal Aviation Administration*, December 1998

[11] Hilburn, T. et al, *Software Assurance Competency Model*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania. (forthcoming)

[12] Watts S. Humphrey, *Managing the Software Process*, Addison Wesley, 1989

[13] Institute for Electrical and Electronic Engineering, *Guide to the Software Engineering Body of Knowledge*, IEEE, 2004

[14] Mead, Nancy R. et al, *Software Assurance Curriculum Project, Volume I: Master of Software Assurance Reference Curriculum* (CMU/SEI-2010-TR-005), Software Engineering Institute, Carnegie Mellon University, 2010.

[15] Mead, Nancy R., Hilburn, Thomas B., and Linger, Richard C., *Software Assurance Curriculum Project, Volume II: Undergraduate Course Outlines* (CMU/SEI-2010-TR-019), Software Engineering Institute, Carnegie Mellon University, 2010.

[16] Mead, Nancy R., Hawthorne, Elizabeth K., and Ardis, Mark, *Software Assurance Curriculum Project, Volume IV: Community College Education* (CMU/SEI-2011-TR-017), Software Engineering Institute, Carnegie Mellon University, 2011.

[17] Moreno, Ana M. et al, "Balancing Software Engineering Education and Industrial Needs," *The Journal of Systems and Software*, vol 85, pp 1607-1620, 2012.

[18] NASA's Systems Engineering Competencies. Washington, DC, USA: US National Aeronautics and Space Administration (NASA), 2009. (http://www.nasa.gov/offices/oce/appel/pm-development/pm_se_competency_framework.html)

[19] Royce, Winston, "Managing the Development of Large Software Systems", Proceedings of IEEE WESCON 26 (August): 1–9

[20] *A Framework for PAB Competency Models*, Professional Advisory Board, IEEE Computer Society, Draft Version, 8-6-12.

[21] *A Competency Model for Software Engineering Practitioners*, Professional Advisory Board, IEEE Computer Society, Draft Version, 8-6-12.

[22] VanLeer, Mary, "Systems Engineering Competency Development," Proceedings of the 5th Annual Conference on Systems Engineering Research, Stevens Institute of Technology, March 14-16, 2007.