Journal of Systemics, Cybernetics and Informatics (2024) 22(5), 62-67
https://doi.org/10.54808/JSCI.22.05.62

Addressing Today's Software Risks Requires an Assurance-Educated Workforce

Carol S. WOODY
Software Engineering Institute
Carnegie Mellon University
4500 Fifth Avenue
Pittsburgh, Pa., 15213-2612, United States

ABSTRACT

There is a significant gap in the current acquisition and engineering workforce's knowledge, skills, and support resources needed to address software and supply chain risk. This gap is widened by two factors: the growing reliance on software to handle system functionality and the exponential increase in cyber attacks. These factors underscore the importance of ensuring that all acquisition software functions as intended and is free from vulnerabilities that can create or contribute to existing cybersecurity issues. However, acquirers, developers, program managers, systems engineers, and decision makers typically lack the knowledge required to create and comply with these requirements. Determining who should be trained and how they should be trained has been an ongoing discussion in the software community for several years. In this paper, we summarize the efforts currently underway to address gaps in workforce knowledge, skills, and support resources based on recent publications and panel discussions held by the Software Assurance Supply Chain (SSCA) forum.

Keywords: Software Risk, Supply Chain Risk, Software Assurance, Assurance Education

1. INTRODUCTION

Today's systems are increasingly software intensive, complex, and reliant on third-party technology. We live in a world of systems of systems linked by software that connects services and hardware, and essentially removes geographic restrictions. By reusing software, systems can be assembled faster and cheaper. However, this approach carries increased risk; many systems engineers do not have software training, and they do not recognize that the processes and practices needed to create and manage software are very different than those used for hardware. One difference is reliability; hardware wears out and must be replaced. However, software does not wear out; it suffers from reliability issues of a different kind. Most software contains vulnerabilities that are difficult to manage directly. Vulnerabilities inherited through the supply chain increase the difficulty of managing all vulnerabilities, and they also magnify the risk of a potential compromise, since the supplier must fix them, and the fix must be applied throughout the supply chain. Suppliers can also propagate malware and ransomware through features that provide automatic updates. As Log4J and SolarWinds have proven, attacks on the software supply chain are increasingly frequent and devastating.

2. GAPS IN ADDRESSING SOFTWARE RISK

The acquisition workforce's knowledge, skills, and support resources must be strengthened, thereby enabling it to address software and supply chain risk. The technology education of acquirers typically does not adequately cover the need for security. Since system functionality increasingly relies on software, and since cyber attacks continue to increase, workforce members must be able to ensure that the software that is part of an acquisition will function according to plan and be free from vulnerabilities that affect its cybersecurity.

The Committee on National Security Systems has defined software assurance as follows [1]:

> *Level of confidence that software is free from vulnerabilities, either intentionally designed into the software or accidentally inserted at anytime during its lifecycle, and that the software functions in the intended manner.*

Workforce members who are responsible for acquiring software are not always aware of the risks involved in acquisition, and they frequently fail to consider the cybersecurity challenges that software represents. Engineers are unaware that the choices they make—coding language, source libraries, software services, and interfaces (to name a few)—can create both immediate risks (i.e., vulnerabilities) and long-term risks (i.e., vendor support). Typically, organizations wait to engage the expertise needed to identify and address these software risks until it is too late (if they address them at all), and these risks can appear anywhere throughout the lifecycle.

One of the last steps in an acquisition is obtaining an authority to operate (ATO), and this is when cybersecurity risk considerations are paramount. However, by the time the acquisition reaches this step, poor choices may have already been made, and fixing them typically requires extensive rework. Acquisition and program management have not recognized that ignoring cybersecurity and software assurance concerns can have major effects on cost and schedule. Many systems engineers who are trained in hardware, do not typically recognize that software must be designed, analyzed, and verified using different processes and practices than those used for hardware. To address this growing software risk, the knowledge of decision makers and participants in system acquisition and engineering must be expanded, but it's not clear who is responsible for this critical component.

How do organizations get the expertise they need to address the growing demand for software that blends in-house code and supply-chain-provided code? The current workforce is unprepared to handle these responsibilities, and the pipeline of future workers has not been educated about software vulnerabilities much less on approaches to addressing them. Complicating matters is that the responsibility for software is widely scattered across many parts of the acquisition and development lifecycle, and collaboration is typically nonexistent among the various workforce members involved.

Organizations make choices about peer reviewing software (and the level of testing required) that ignore cybersecurity risks. Software developers and development pipeline creators make cybersecurity choices based on their tooling selections and the built-in restrictions. Contractors and subcontractors make choices about coding language, libraries, reuse, third-party products, open source software, and related software policies and practices that affect software assurance and its cybersecurity protection. Once these choices are "baked" into the system, an ongoing relationship continues with the supplier that must be maintained or replaced. Policies and practices must be established to address the need for continuous monitoring for new vulnerabilities to evaluate and prepare for potential impacts. Few members of the current and incoming workforce are prepared to handle these activities and make decisions effectively.

3. CLOSING THE GAP

The Software Engineering Institute (SEI) conducts research on the acquisition and development lifecycle that has revealed opportunities for improving both cybersecurity and software assurance. (See Figure 1.) New methods, processes, and practices should be integrated into all stages of the lifecycle, initially to predict effective results during design and later to confirm the desired results during testing, verification, and validation. The SEI published the *Acquisition Security Framework* (ASF), which describes these new methods, processes, and practices [2].as The ASF is an assembly of the practices needed to effectively address software assurance across the lifecycle. It includes 51 goals and 334 practices, which are spread across the following six practice areas:

- Program Management
- Engineering Lifecycle
- Supplier Dependency Management
- Support
- Assessment and Compliance
- Process Management

These practice areas affect a wide range of workforce members in systems and software development. However, too frequently, organizations neglect to assign resources to address the tasks in these practice areas because leadership does not understand what is needed to achieve effective cybersecurity and software assurance risk mitigation. Furthermore, most organizations lack workforce members with the requisite skills to perform these activities effectively.

New processes and practices could be integrated into all stages of the lifecycle that will predict effective results during design and confirm the desired results during testing, verification, and validation. However, adoption of these new steps has been slow because it is very difficult to demonstrate a return on investment. Cybersecurity has focused on controls and not measurements, and current processes and practices allow organizations to successfully address compliance mandates without making the desired improvements. New compliance mandates *are* coming (e.g., Cybersecurity Maturity Model Certification [CMMC]); however, most organizations do not have the resources to address them effectively.

Early efforts to educate the incoming workforce have included developing software assurance curricula approved by the Institute of Electrical and Electronics Engineers (IEEE) and the Association for Computing Machinery (ACM) [3] and courses that supplement existing cybersecurity engineering. However, implementing this new approach was limited since this material required different skills to teach and there was limited course-ready material. Several early adopters of this approach integrated material into higher education, but these modest efforts had little impact on the large number of students moving into careers such as software development. Furthermore, many of these workforce participants come from community colleges, not four-year colleges or graduate programs, where the curriculum is more likely to include software assurance.

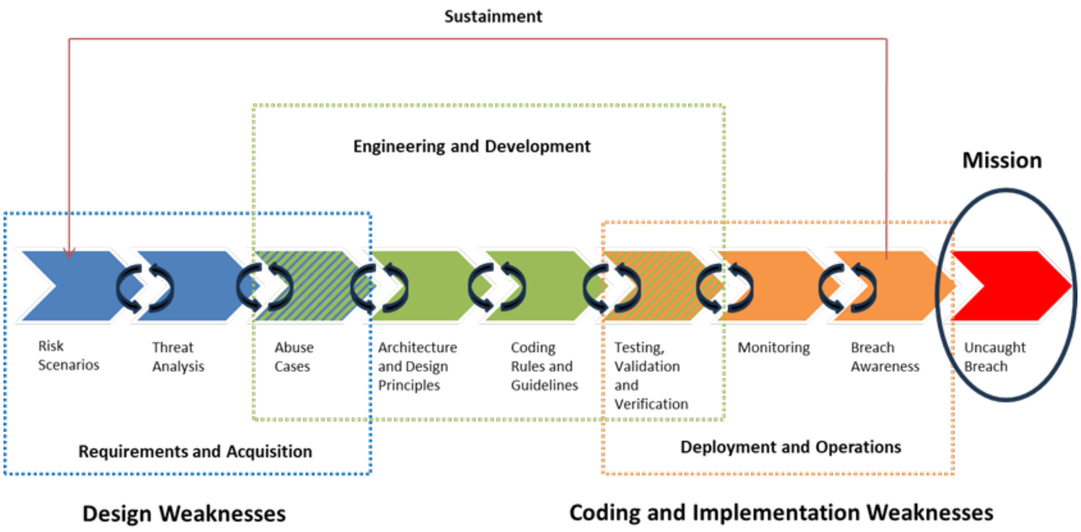
Sustainment
Engineering and Development
Mission
Risk Scenarios
Threat Analysis
Abuse Cases
Architecture and Design Principles
Coding Rules and Guidelines
Testing, Validation and Verification
Monitoring
Breach Awareness
Uncaught Breach
Requirements and Acquisition
Deployment and Operations
Design Weaknesses
Coding and Implementation Weaknesses

Figure 1: Activities That Address Defect and Vulnerability Identification and Mitigation

Program managers focus on cost and schedule. They are tasked to deliver a product that addresses the functionality specified in the contract. The product will also have other characteristics that are not as clearly specified, sometimes referred to as non-functional characteristics because they are not directly related to performance. These characteristics include safety, security, reliability, maintainability, and modularity, which are all part of how the product is designed and built. These characteristics are influenced by choices made in engineering and design. Standards, compliance mandates, and engineering expertise all influence engineering choices. Cybersecurity focuses on the controls placed on the operational processing of the system; software assurance, with its emphasis on removing vulnerabilities in the system, focuses only on the construction.

The responsibility to deliver safety, security, reliability, and other features within a product belongs to engineering, but many engineers do not understand these features. Merely conducting static analysis on the code, fixing problems the tools identify as high priority, and inserting a raft of controls that attempt to limit access to various parts of the system are insufficient. Tools are not tuned to prioritize vulnerabilities based on the risk they represent to an individual system's design. Likewise, software in interconnected systems can have unexpected behaviors that could allow system controls to be bypassed. Threats must be evaluated from both a systems and software processing perspective to confirm that appropriate constraints are in place to ensure expected behaviors. Many design and architecture choices that are inappropriately applied will leave a system vulnerable to cyber attacks. Not all engineers and developers have the level of expertise needed to understand the complexities of highly connected software-intensive systems.

The extensive inclusion of third-party code (e.g., code language libraries, commercial off-the-shelf [COTS] software, government off-the-shelf [GOTS] software, open source software) and third-party services (e.g., the cloud) invites a broader range of software into the system that must meet the needed level of software assurance. However, since this third-party code was developed to meet different requirements, the likelihood it will meet the expectations and requirements for the system to be acquired is highly uncertain.

The lack of software assurance rigor in engineering and acquisition decision making has led to widespread problems with the availability of needed workforce members and gaps in their knowledge, skills, and support resources, which increases the "attackability" of systems. Too frequently, we find that the workforce members tasked to make these decisions have no understanding of how software can be compromised to put system functionality at risk. The training needed to enable workforce members to address software assurance is not part of the standard training for computer science, systems engineering, or software engineering, and it is not a requirement for program management. Addressing these gaps in software assurance training must include addressing the lack of software assurance knowledge among the key decision makers who handle acquisition and development.

In May 2018, a two-day workshop convened at the Software Supply Chain Assurance (SSCA) Forum. The workshop was co-led by the U.S. National Institute of Standards and Technology (NIST), the U.S. Department of Homeland Security (DHS), the U.S. Department of Defense (DoD), and the U.S. Government Services Agency (GSA). The purpose of the workshop was to address important questions related to education, training, and certifications for software assurance and supply chain risk management. The Idea Group, Inc. (IGI) published the details of the forum [4]. The 118 attendees, who represented industry, government, and academia, met to address the following concerns:

- What are the challenges facing industry, academia, and government organizations in this area?
- Who needs education or training?
- What needs to be taught?
- What strategies do or do not work?

The workshop confirmed that few institutions teach courses relevant to software assurance. The International Information System Security Certification Consortium (ISC_2) offers specialized certifications that focus on cybersecurity, but they primarily focus on systems and software in the operational environment. Supply chain considerations primarily focus on hardware.

Since 2018, the scope and impact of supply chain issues has grown exponentially, including SolarWinds (2020), Log4j (2021), Medibank (2022), MOVEit (2023), and CrowdStrike (2024). An internet search on any one of these topics returns extensive details about what happened and why. However, for any organization building and using software, how to address these risks remains largely undecided. Builders and maintainers in this complex environment of highly interconnected systems and software must understand how to identify and address cybersecurity risk, supply chain risk, and software assurance. There are tools that can help, but they require effective processes and practices integrated into the enterprise management of acquisition and development to produce effective, repeatable results.

In January 2024, I facilitated a panel for the SSCA Forum to "Establish the Demand Signal for Good Software Assurance." In this panel, participants shared their experience by addressing the following questions:

- What has been your motivation for addressing the software assurance challenge?
- If you were starting your career planning today, what would you want to learn about software assurance to position you to be an exceptional job candidate?
- Where would you want to be able to learn this (school, on-the-job training, online, ChatGPT)?
- What practices and environments do workplaces need for these educated workers to have an impact?
- How might you evaluate job candidates for this capability?

Government participants shared how they enhanced software assurance through improvements to the following:

- policy
- requirements that focus on delivering software with fewer vulnerabilities
- training courses they developed that enhance acquisition members' understanding of software assurance and challenge developers to learn how an attacker can leverage software to successfully attack their systems

Participants emphasized expanding acquisition expertise to ensure that data is protected in services (e.g., cloud services) just as it would be on premises, and they recognized the supply chain as an immediate risk to be addressed.

Participants from academia described how they enhanced existing courses to include key concepts and projects for students to build an understanding of software risk. However, greater effort is needed to ensure that every developer understands what a vulnerability is, what tools are available for finding them, and how they can avoid inserting them in their code.

In May 2024 at the SSCA Forum, I moderated a panel titled "Positioning for Software Assurance Success: Practices, Tools & Technology, Knowledge, & Skills." Participants from government and industry shared their experiences addressing software assurance using what we have come to see as the following three key areas for success:

1. In **Software Assurance Practices**, participants addressed the following questions:

    - Does your domain have adequate software assurance practices?
    - Does your organization use them?
    - What's missing?

2. In **Software Assurance Tools & Technology**, panel members addressed the following questions:

    - Do effective software assurance tools exist for your domain?
    - Does your organization use them effectively?

3. In **Software Assurance Knowledge & Skills**, they addressed the following questions:

    - What three software assurance knowledge and skill sets do technologists in your domain need most?
    - Where can they get adequate education and training in them?

Audience members were widely scattered across the software and systems lifecycle. Many indicated that current solutions were not sufficient to address the problems, which leave organizations at risk. The expertise to understand the risk and promote effective solutions is not widely available. Several expressed frustration that earlier efforts to assemble useful expertise have not been maintained, such as the report titled *State-of-the-Art Resources (SOAR) for Software Vulnerability Detection, Test, and Evaluation* [5]. This information is helpful for practitioners, learners, and instructors. Funding for creating new capabilities appears to be accessible, but the ongoing support to maintain funding is more difficult.

Panel members emphasized that organizations need to establish consistent processes and practices for software assurance that apply across acquisition, engineering, development, cybersecurity, safety, and other areas so that information sharing, problem resolution, and consistent results can be delivered. Software risks need to be tracked in the same way that costs and schedule risks are monitored and managed. Too many choices are made by engineers and developers with little understanding of their impacts. There is much more guidance available than anyone has the time or interest to read. Ensuring that the guidance works within the organization requires a culture that promotes quality and reduced risk for software instead of speedy delivery. Incentives for software development are currently backwards.

As more third-party software integrates with open source software, considerations related to open source software become critical. With an emphasis on speedy delivery, open source software is seen as a bonanza of free material to be mined. Assembling and monitoring software bills of material (SBOMs) [6] will be a growing method for recognizing and mitigating vulnerabilities, no matter the software's source. The White House Executive Order issued in May 2021 [7] emphasizes collecting SBOM information. However, to be successful, organizations will need to use this data to effectively manage the software they build and buy. Current tooling is inadequate because it requires highly skilled experts to apply the Executive Order effectively, and much of the available workforce lacks those skills. Leaders must build their own knowledge of the risks involved; baseline their organization's current approach; and augment gaps with policy, processes, practices, standards, and training. There is no easy fix.

A good place to start is identifying what to measure and monitor to establish software assurance results. Simply counting defects or vulnerabilities is an unending cycle. Instead, organizations should look for ways to integrate measurement into existing lifecycle activities. In the current state of the practice, it's easy to collect vast amounts of data related to cybersecurity and software assurance, but it is extremely challenging to structure that data to support decision making [8]. Selecting the appropriate metrics for a specific acquisition and development project requires identifying what is possible and what decisions need to be made [9]. In many cases, needed measures are too costly, so close substitutes must be used.

4. SUMMARY

Addressing software assurance for systems that are increasingly dependent on software, highly interconnected, and reliant on supply chain components is an extremely complex task. Workforce members who are leading projects to build the parts and pieces of these complex systems must work together to ensure the delivered system can minimize the risk to their missions. This collaborative work cannot be left to chance. Expertise in software assurance, cybersecurity risk management, and software supply chain risk management, which includes open source management, must be highly integrated into the decision making, design, development, and management of every aspect of the lifecycle. This journey is not a point-in-time activity. Bringing the right resources to support this journey is critical to success. It will include training the current workforce and better preparing the future workforce to deliver the systems we need with the level of assurance we require.

5. ACKNOWLEDGEMENTS

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

DM24-1011

6. REFERENCES

[1] Committee on National Security Systems. National Information Assurance (AI) Glossary. *CNSS Instruction No. 4009. Committee on National Security Systems*. April 26, 2010. https://www.dni.gov/files/NCSC/documents/nittf/CNSSI-4009_National_Information_Assurance.pdf

[2] Alberts, Christopher; Bandor, Michael; Wallen, Charles; & Woody, Carol. *Acquisition Security Framework (ASF): Managing Systems Cybersecurity Risk (Expanded Set of Practices)*. CMU/SEI-2023-TN-004. Software Engineering Institute. 2023. https://insights.sei.cmu.edu/library/acquisition-security-framework-asf-managing-systems-cybersecurity-risk-expanded-set-of-practices/

[3] Software Engineering Institute. Software Assurance Curricula. *Software Engineering Institute*. August 6, 2024 [accessed]. https://www.sei.cmu.edu/education-outreach/curricula/software-assurance/index.cfm

[4] Boyens, B., "Opinions of the Software and Supply Chain Assurance Forum on Education, Training, and Certifications." *International Journal of Systems and Software Security and Protection*. Volume 9. Issue 2. April-June 2018. https://doi.org/10.4018/978-1-6684-3554-0.ch009

[5] Fong, E. Kenneth Hong; Wheeler, David A.; & Henninger, Amy E. *State-of-the-Art Resources (SOAR) for Software Vulnerability Detection, Test, and Evaluation 2016*. IDA P-8005. Institute for Defense Analysis. November 2016. https://ida.org/research-and-publications/publications/all/s/st/stateoftheart-resources-soar-for-software-vulnerability-detection-test-and-evaluation-2016

[6] United States Department of Commerce. *The Minimum Essential Elements of a Software Bill of Materials*. United States Department of Commerce. July 12, 2021. https://www.ntia.doc.gov/files/ntia/publications/sbom_minimum_elements_report.pdf

[7] White House. Executive Order on Improving the Nation's Cybersecurity. *White House*. May 12, 2021. https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/

[8] Mead, Nancy; Woody, Carol; & Hissam, Scott. The Measurement Challenges in Software Assurance and Supply Chain Risk Management [blog post]. *Software Engineering Institute*. December 2023. https://insights.sei.cmu.edu/library/measurement-challenges-in-sw-assurance-and-scrm-white-paper/

[9] Woody, Carol; Ellison, Robert; & Ryan, Charles. *Exploring the Use of Metrics for Software Assurance*. CMU/SEI-2018-TN-004. Software Engineering Institute. 2019. https://insights.sei.cmu.edu/library/exploring-the-use-of-metrics-for-software-assurance/