Ensuring Software Assurance Process Maturity

Edmund Wotring III, Information Security Solutions, LLC
Sammy Migues, Cigital, Inc.

Abstract. All organizations–government and commercial alike–share an interest in minimizing their software vulnerabilities and, consequently, in maturing their software assurance capabilities. Successful software assurance initiatives require organizations to perform risk management activities throughout the software lifecycle. These activities help to ensure organizations can meet software assurance goals, including those related to reliability, resilience, security, and compliance. The Software Assurance (SwA) Checklist for Software Supply Chain Risk Management (hereafter referred to as the SwA Checklist) serves as a framework to help organizations establish a baseline of their risk management practices and select maturity model components to better meet evolving assurance goals.

Introduction

Software assurance is the level of confidence that software is free from vulnerabilities, whether intentionally designed into the software or accidentally inserted at any time during its lifecycle, and that it functions in the intended manner.[1] Once an organization becomes aware of the need to meet software assurance goals, the next step is to assess its current development and procurement activities and practices. Such an analysis requires at least two things. The first is a repeatable and objective assessment process. The second is a clear benchmark or target that represents a suitable level of risk management given the nature of the organization and the software's mission. Performing this assessment periodically provides an ongoing understanding of the maturity of respective software assurance capabilities.

Choosing a methodology for appraising an organization's ability to meet software assurance goals may seem overwhelming because there are several maturity models available, each with their own focus and level of granularity. For an organization that may be new to the area of software assurance, it can be a challenge to simply find good sources of guidance, much less understand which parts of each model are best suited for its environment and supply chain. Although finding the right maturity model may seem challenging, organizations should not wait for an authority to mandate a software assurance initiative. Such mandates are typically intended to be "one-size-fits-all" and offer limited flexibility. Organizations are best served by tailoring a software assurance strategy to their own supply chains.

Selecting the best maturity model, or model components, for a particular organization to begin addressing assurance goals may also present a time-consuming learning curve. In order to facilitate an understanding of how multiple maturity models address similar assurance goals, the authors created a model-agnostic framework as part of participation in the SwA Forum Processes and Practices (P&P) Working Group (WG), which is co-sponsored by organizations with DHS, DoD, and the National Institute for Standards and Technology. This analysis involved mapping maturity models, and their respective practices, within the framework. The agreement among the models provides a valuable reference. This framework evolved into the SwA Checklist, which serves as a model-agnostic harmonized view of software assurance guidance.

The SwA Checklist can help organizations begin a dialogue amongst the entities in the supply chain that influence and/or support the software throughout the lifecycle. Using the checklist to characterize each of the organizations in a given supply chain provides extraordinary insight into the credibility or trust deserved by a given piece of software. By leveraging this insight, organizations can verify implicit assumptions that certain practices are taking place and align their activities with assurance goals to mitigate risks within their supply chains. Organizations can also use the checklist to organize evidence for assurance claims while assessing all of its practices as it performs the activities necessary to complete its baseline. Finally, organizations can use the baseline to engage their senior leadership regarding the areas in which resources are needed to meet assurance goals based upon guidance from the mapped models.

The SwA Checklist provides a consolidated view of current software assurance best practices in the context of an organized SwA initiative. The checklist is currently implemented as a "hot linked" Microsoft Excel spreadsheet that provides a cross-reference of goals and practices with side-by-side mappings to several publicly available maturity models. Organizations can use the mappings to identify where the maturity models agree and diverge, and use this consolidated format to select model components best suited to their environments.

Once an organization establishes its assurance goals, selects a maturity model (or model components), and captures its baseline, it can then establish an improvement plan for achieving software assurance goals as it develops and/or acquires secure software. Working with its direct customers (downstream in the supply chain) and suppliers (upstream in the supply chain) to improve software assurance will have a large multiplier effect as the approach spreads to other organizations.

Intended Use

The intended users of the SwA Checklist are organizations that currently are or soon will be acquiring or developing software. Organizations may have many options when developing or acquiring software from various sources. Although vendors and developers may offer software that meets specified functional requirements and provides myriad features, these offers are inconsequential if the data and functions are not protected. Developers and acquirers must give significant consideration to the ability of the software to reliably function and protect data and processes over the life of the product. Organizations can use the SwA Checklist to guide their own development or to evaluate vendor capabilities. Organizations can use the baselines they establish to facilitate an understanding of similar assurance goals and practices among several freely available maturity models, which can help guide the selection of the most appropriate model components.

Design of the SwA Checklist

The SwA Checklist is available at no cost at <https://buildsecurityin.us-cert.gov/swa/proself_assm.html>. The SwA Checklist is currently being vetted and we request your feedback based upon practical use within the field. A feedback form is available at the same URL above. The authors designed the checklist to be understandable by users with various levels of SwA experience (readers are invited to download a copy now and review it while reading this section).

The SwA Checklist contains multiple tabs/worksheets including the following: Intro, SwA Checklist, Sources, BSIMM, CMMI-ACQ, OSAMM, PRM, and RMM. The "Intro" tab serves as the introductory section that also provides pointers to each of the included models. The "SwA Checklist" tab provides the information that enables users to perform their analysis. Content from the included models is organized into five domains: Governance, Knowledge, Verification, Deployment, and Supplier Management. This categorization helps to harmonize terminology and makes it easy for the user to locate specific guidance. Within each domain are three categories containing a short, high-level goal and a set of three corresponding practices. There is a "Status" cell under each practice. Users can click on the cell to open a pull-down menu with pre-defined responses to input their organization's implementation status for each practice. The range of possible status levels in the pull-down menus includes the following:

- **Unknown**
- **Not Applicable**
- **Not Started**
- **Partially Implemented Internally**
- **Partially Implemented by Supplier(s)**
- **Partially Implemented Internally and by Supplier(s)**
- **Fully Implemented Internally**
- **Fully Implemented by Supplier(s)**
- **Fully Implemented Internally and by Supplier(s)**

It is the combination of the status of each practice that will help an organization understand its ability to execute on software assurance activities in development and acquisition.

SwA Tools Relationship

Another tool that is mapped to multiple maturity models, the SwA Self-Assessment, is also available on the same webpage on the DHS SwA Community Resources and Information Clearinghouse website. The SwA Checklist and the SwA Self-Assessment are resources made available from the SwA Forum. The tools provide alternative views on similar assurance process frameworks whose shared objective is software improvement. It is in an organization's best interest to try both approaches and use the one that works best for its own environment. No matter which tool users select, it is important to remember the ultimate goal is producing and delivering rugged software.

The implementation status options vary based upon the degree to which the practice is implemented (i.e., not started, partially implemented, or fully implemented) and the party responsible for each practice (i.e., internally, by the supplier, or by both). The two other responses included in the pull-down menu are "Unknown" and "Not Applicable." The user should follow up on any response marked with either of these statuses. Organizations should mark a practice "Unknown" if it is unknown whether someone is performing the practice or who is responsible for performing it. Such a practice is almost certainly an area of increased risk and requires further investigation. Likewise, if a practice is marked as "Not Applicable," the user should obtain justification for selection of that status. Supply chain partners must understand the environment in which the software will be deployed and meet the end customers' assurance needs even if those needs are not explicitly stated. When assurance goals are analyzed from such derived requirements, certain practices may reveal themselves as applicable. Thoroughly investigating the status of each practice is a valuable due diligence exercise that may result in the user discovering that certain practices actually are applicable or that practices are already being performed as part of other related practices.

By performing the analysis required to assign a status to each practice, the user gains a greater understanding of their overall supply chain and establishes an assurance baseline. This understanding will enable more productive dialogue among all supply chain parties and will foster better understanding of where risk is introduced during acquisition or development of software.

Maturity Model Mappings

The third tab of the spreadsheet, Sources, includes all the same goals and practices from the SwA Checklist tab. Table 1 contains a portion of this view. The Sources tab also includes mappings for each practice to several maturity models, described in the sidebar to this paper on page No. 32 titled Maturity Models (Maturity Models Mapped within the SwA Checklist). All mappings are hyperlinked to other tabs in the spreadsheet. Clicking on a hyperlinked mapping will take the user to the related section on the tab for the corresponding maturity model. The user can return to the Sources tab by clicking on the hyperlinks in column A of any of the maturity model tabs.

There are several benefits to viewing the mappings for each practice in the SwA Checklist side-by-side in the Sources tab. The mappings help the user to see how the maturity models agree and diverge on each of the related practices. Since each model has its own particular focus, viewing the relationships

Maturity Models

There are several freely available maturity models that focus on securing software. Each has its own focus and level of granularity. The publicly available maturity models mapped in the Sources tab of the SwA Checklist include:

- Building Security In Maturity Model version 2 <**http://www.bsimm.com**>
- Carnegie Mellon University SEI CMMI® for Acquisitions, version 1.2 <**http://www.sei.cmu.edu/cmmi/index.cfm**>
- Open Web Application Security Project Open Software Assurance Maturity Model version 1.0 <**http://www.opensamm.org**>
- Software Assurance Forum Processes and Practices Working Group Assurance Process Reference Model, September 2010 <**https://buildsecurityin.uscert.gov/swa/downloads/20100922_PRM_Practice_List.pdf**>
- Carnegie Mellon University/CERT Resiliency Management Model, version 1.0 <**http://www.cert.org/resilience/rmm.html**>

The authors performed a model-agnostic analysis to determine how these maturity models help organizations address assurance goals and practices and to determine where the models converge and diverge. This analysis of the mappings between the models revealed a high degree of agreement. Organizations can use the checklist to determine process improvement opportunities and establish a baseline from which to benchmark their capabilities. More information on the maturity models analyzed and included in the SwA Checklist is available at <https://buildsecurityin.us-cert.gov/swa/proself_assm.html>.

among them provides a context from which the user can better understand the assurance goals and practices. The user will also see how various models address similar goals and practices. This will help the user begin selecting a maturity model that will be of most use to their particular software assurance needs.

Table 1: Sources Tab Snapshot

	Governance		
	Strategy & Metrics	Policy & Compliance	Training & Guidance
Practices	Establishes Security Plan; communicates and provides training for the plan	Identifies and monitors relevant compliance drivers	Conducts security awareness training regularly
BSIMM	SM1.1	CP1.1	T1.1
	-	CP1.2	T3.4
CMMI-ACQ	PP SG2 – SG3	OPF SG1	OT SG2
	-	-	-
OSAMM	SM1B	PC1A	EG1A
	-	PC1B	-
PRM	SG 2.1	SG 3.1	SG 1.3
	SG 1.3	-	-
RMM	RTSE: SG2 – SG3	COMP: SG2	OTA: SG1 – SG2
	MON: SG1	MON: SG1 – SG2	-

Appraisal Considerations

When performing an appraisal using the SwA Checklist, it is important that the user adapt the checklist to the processes being performed and the structure of their organization's supply chain. Users may determine that they implement a different practice that also supports an assurance goal in the checklist. This is typical since not all organizations employ the same practices despite desiring roughly the same assurance goals. Users may also perform an evaluation of a supplier or a division of an organization that only manages a portion of the processes in the overall supply chain. In this case, it is likely that not all the goals and practices within the checklist will apply to this specific supplier or division. Users should leverage the SwA Checklist to determine whether they are taking a comprehensive approach to produce rugged software throughout the entire supply chain. This approach may require evaluating multiple suppliers, divisions, and other entities to comprehensively manage risks and to ensure supply chain partners meet assurance goals.

The mappings of the models in the Sources tab provide valuable reference and context as users complete a baseline. As users become more aware of how the models address similar goals and practices, they may begin to find currently unimplemented model components that are useful for their environments and specific assurance needs. The models referenced within the checklist are designed with varying levels of granularity ranging from high-level control objectives to lower level controls. Each of these perspectives may provide insight into addressing the assurance challenges in various supply chain environments.

Baseline Summary

After users establish a baseline, a summary displays at the bottom of the SwA Checklist tab. This summary depicts a count of each category of implementation status and is highlighted in a conditional formatting color scheme according to the following:

"Not Applicable" practices – Grey
"Unknown" and "Not Started" practices – Red
"Partially Implemented" practices – Yellow
"Fully Implemented" practices – Green

This system provides an easy-to-view dashboard for an organization's overall implementation of practices.

The color-coded system provides a way to quickly assimilate data contained within the user-created baseline. Although the system uses stoplight colors, improvement efforts should not focus solely on the "reds" and "yellows." A practice highlighted in green does not necessarily satisfy the organization's assurance goals or adequately mitigate risks. Further, a practice highlighted in green is one that is being performed, not necessarily one that is required. Organizations must analyze the entire checklist to determine if the correct entity performs each practice correctly and to a sufficient extent, and if each practice is actually mitigating risks according to the organization's assurance goals. Only after determining these factors can the organization outline a plan to effectively and efficiently improve its software assurance capabilities.

Common Appraisal Challenges

The most common issue users face when creating a baseline pertains to practices for which the status is "Unknown." In these instances, the best approach may be to document the process flow surrounding the practice. It is helpful to coordinate with the parties involved in processes surrounding the practice to determine the degree to which the process is implemented. Determining responsibility for each practice is another common issue faced by users. Appraisers should diligently clarify accountability and responsibility during their analyses. The third frequently arising issue is tracking execution of software assurance activities and ensuring suppliers and acquirers do them consistently and effectively. Even when users know what practices are implemented and who is responsible for them, they may be unaware how well they are implemented. Lastly, if users know a practice is implemented, who is responsible for its implementation, and whether it is executed correctly, they still may not know whether it is effectively reducing risk and should be continued.

Even though the practices marked as "Fully Implemented" on the checklist will register as green, this does not necessarily mean they represent money (or resources) well spent. It is important for organizations to select components from the source models to improve the implementation of practices specifically required to meet assurance goals and to ensure their satisfactory completion. It is important to measure not only the assurance activities, but also the software lifecycle artifacts (e.g., code) to ensure both are improving. Overall, organizations should determine the model components that help them accomplish a coherent and cohesive set of activities that accomplish organizational goals based upon business objectives and risk appetite.

Conclusion

Establishing an implementation baseline of the practices within an organization's supply chain will foster a better understanding of its true capability to develop, acquire, and deploy secure software. Using the checklist, an organization may identify opportunities for improvement and begin to create a plan to address improvement areas by selecting model components from the mapped maturity models. The more robust the processes are surrounding software lifecycle processes, the more likely an organization will develop and acquire truly rugged software. The SwA Forum P&P WG plans to periodically update the SwA Checklist to ensure it aligns with updated versions of the models mapped in the Sources tab and to incorporate other models into this mapping in the future.

Acknowledgements

This work is funded in part by the DHS Software Assurance Program. Many colleagues and members of the Software Assurance community provided valuable feedback on the checklist and this article including: Joe Jarzombek, DHS; Don Davidson, OASD-NII / DoD CIO; Michele Moss, Booz Allen Hamilton; Lisa R. Young, CERT; Walter Houser, SRA; Doug Wilson, Mandiant; Rama Moorthy, Hatha Systems; and Dr. Robin A. Gandhi, Nebraska University Center for Information Assurance.

ABOUT THE AUTHORS

Edmund Wotring III is a Senior Security Engineer with Information Security Solutions, LLC. He previously supported various federal government clients with security compliance and process improvement initiatives. He has advised senior leadership to ensure compliance processes facilitate effective security. He currently supports the Department of Homeland Security National Cyber Security Division's Software Assurance program.

E-mail: ed.wotring@informationsecuritysolutionsllc.com

Sammy Migues is a Principal and Director of Knowledge Management at Cigital. He has nearly 30 years experience performing security research and providing practical solutions to government and commercial customers. He is currently working on expanding the BSIMM research, smart grid security demonstration projects, new methods of software security training, and helping organizations start or grow software security initiatives.

E-mail: smigues@cigital.com

NOTES

1. Committee on National Security Systems (26 April 2010). CNSS Instruction No. 4009. National Information Assurance (IA) Glossary. [Accessed 02 Nov 2010]. Available from: <http://www.cnss.gov/Assets/pdf/cnssi_4009.pdf>.

Disclaimer:

® CMMI is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.