Secure Software Framework Comparisons

Megan Leonard

ISEC 475

6/11/2023

When it comes to software frameworks, there are those that focus on the development through each step while others that focus on security called, security software development frameworks. These frameworks each have their own strengths and weaknesses while differing in their focus and what part they see as the most important.

One of these secure frameworks is CMMI, capability maturity model integration. This was originally created for software engineering and is a behavioral model that focuses on performance. The behavioral model helps create a better understanding of what works for the organization and gives detailed assessments. It allows for a greater insight into processes that have been around longer.

The problem is that it is not applicable to every organization so while it could help one, it could harm another. CMMI also does not meet the security standards of every business so it may be able to meet the standards of one but would fall short with another. Then there is the problem of compatibility where CMMI may not be compatible with other frameworks and processes.

Another secure framework is the BSIMM which means building security in maturity model. This one focuses on the different practices and finding the common ground between them. BSIMM helps with observing what others are doing and comparing your business to another. This would work well with another framework as it focuses more on comparison and observation.

The downside for BSIMM is that it is mostly just comparing so implementing it should be done with another or a preexisting build. It does not provide the building structure itself and relies on information it takes from others and the commonality between them all.

The next one is the Microsoft SDL, security development lifecycle. This is a framework that was created by Microsoft and is standards that their company adheres to. It is helpful as it splits its structure into seven major categories where two are used for adding security support. It is well known and used by many as it works on securing software.

The problem is that Microsoft SDL is quite rigorous and made with large companies in mind so it may hinder rather than help smaller companies. There is also the fact that each stage has mandatory checks, so the process slows down.

OWASP SAMM, software assurance maturity model, is the final framework we are looking at. This is an open-source framework which looks at a company’s existing security practices and creates a balance between that and their security assurance. It helps with creating a strategy specific to the organization instead of making the organization conform to its standards.

The downside of OWASP SAMM is that you need to have a preexisting security practice and at least a basic understanding of frameworks. It helps with existing ones, so it is not as well used on its own as when it is with another.

Each framework has its own strengths and weaknesses meaning that they are best implemented in different scenarios. OWASP SAMM is best used when there is an existing framework as it can help build on it. Microsoft SDL works well on its own for large organizations. BSIMM is similar to OWASP SAMM with how it works best with a preexisting framework but unlike SAMM, it only compares structures not build off of them. CMMI is another standalone framework like Microsoft SDL but could be used with smaller companies. It also cannot work with other frameworks like BSIMM and SAMM and does not hold up too many standards like Microsoft SDL does. As they all have their or use times, each is built for different scenarios so it not many would fit for the same project.

Citation:

Bart De Win, AbstractDevelopment processes for software construction are common knowledge and mainstream practice in most development organizations. Unfortunately, Howard, M., McGraw, G., Nuseibeh, B., Davenport, T., Jacobson, I., & Beck, K. (2008, February 15). *On the Secure Software Development Process: Clasp, SDL and touchpoints compared*. Information and Software Technology. <https://www.sciencedirect.com/science/article/abs/pii/S0950584908000281>

Gluckd. (n.d.). *Microsoft Security Development Lifecycle (SDL) - microsoft service assurance*. Microsoft Service Assurance | Microsoft Learn. <https://learn.microsoft.com/en-us/compliance/assurance/assurance-microsoft-security-development-lifecycle>

H, A. (2023, March 29). *BSIMM vs Samm: Which is the better AppSec programme*. Codific. <https://codific.com/bsimm-vs-samm/>

*Owasp Samm*. OWASP SAMM | OWASP Foundation. (n.d.). <https://owasp.org/www-project-samm/>

Pivot Point Security. (2022, June 9). *What is Owasp Samm and why should we (as an org that develops software) care?*. Pivot Point Security. <https://www.pivotpointsecurity.com/what-is-owasp-samm-and-why-should-we-care/>

School of Informatics | The University of Edinburgh. (n.d.). <https://www.inf.ed.ac.uk/teaching/courses/sp/2015/lecs/BSIMM6.pdf>

Team, D. A. (2022, September 16). *Pros and cons of the CMMI model*. Digital Adoption. <https://www.digital-adoption.com/cmmi/>

*What is Cmmi? A model for optimizing development processes*. CIO. (2021, June 1). <https://www.cio.com/article/274530/process-improvement-capability-maturity-model-integration-cmmi-definition-and-solutions.html>