November 26 2013

WP002

The Case for an Open and Evolving Software Assurance Framework

Miron Livny, SWAMP Director and CTO
Morgridge Institute for Research
Barton Miller, SWAMP Chief Scientist
University of Wisconsin-Madison

This document explains the theory behind the SWAMP framework to meet the changing needs and expectations of users in the SwA community.

Citation Information

MLA

Livny, Miron, and Miller, Barton. “The Case for an Open and Evolving Software Assurance Framework.” *Continuous Software Assurance Marketplace*, 26 Nov. 2013. Web. <https://www.swampinabox.org/doc/SWAMP-WP002-Framework.pdf>.

APA

Livny, M., & Miller, B. (2013, November 26). The Case for an Open and Evolving Software Assurance Framework [PDF file]. *Continuous Software Assurance Marketplace*. Retrieved from https://www.swampinabox.org/doc/SWAMP-WP002-Framework.pdf

The Case for an Open and Evolving Software Assurance Framework

Evaluating the assurance of software artifacts involves solving a broad spectrum of problems. Each of these problems constitutes a stage in an end-to-end workflow, where each stage is realized by a collection of tools that addresses the specific tasks of that stage. Extending the impact and expanding the reach of software assurance technologies requires a model that captures the different stages in these workflows and a framework that embodies it. Such a framework will cover the stages of code analysis, result normalization and labeling, result merging and integration, visualization, result evaluation and annotation, and risk assessment. The model and framework cannot be static, as experience and innovation will evolve them well past the limits of today's technologies. Our understanding of software assurance challenges and novel methodologies will continue to grow as new technologies are developed.

One of the key benefits of a flexible and adaptive framework is the ability to compose a variety of tools to produce a workflow that is tailored to the specific needs of a software assurance task. Operating on a chain of well-defined intermediate results, these "best of breed" tools will join forces to deliver effective assurance capabilities to the end user. The value and power of such frameworks have been effectively demonstrated in a variety of technology areas as they also facilitate sharing within and across organizations. Easy authoring, exchange and adaptation of workflows facilitate the development and adoption of best practices throughout the community.

Another key impact of such a framework is the ease in which new technologies are adopted. By offering the "glue" needed to include a technology in a workflow, the framework expedites the "the time to impact" of novel technologies. It minimizes the burden placed on the technology developers who, in most cases, do not have the means to develop the required utilities needed to make their technologies accessible to end users.

The need for an open and flexible software assurance (SwA) framework has guided the design and development of the Department of Homeland Security Science and Technology Directorate's recent initiative, the Software Assurance Marketplace (SWAMP). As a technology-neutral entity, the SWAMP is uniquely positioned to define, implement and evolve such a framework and to make it available to the SwA community. As the SWAMP adds SwA capabilities and brings in new partners, it is expanding its coverage of the framework and defining its new aspects.

Software developers looking to increase the assurance in their code can come to the SWAMP, choose the tools and technologies that best fit their needs, and compose them into complete workflows. The continuous assurance capabilities of the SWAMP require the repetitive execution of these workflows as the software and/or the tools change. Developers benefit from the variety of tools available at each stage and the experiences of other developers, as embodied in the stored workflows, who have used the SWAMP to solve similar problems. Developers can also bring their own tools to the SWAMP to

experiment with new technologies and methodologies. As developers gain more experience with this global view of the SwA process, they can add more tools to their workflow, and expand their coverage of the problem space.

In recent years we have seen several organizations take steps to support a more flexible and composible approach to the software assurance process. These groups include Secure Decisions's CodeDX and Denim Group's ThreadFix tools, which merge and visualize results from multiple code analysis tools. These groups also include KDM Analytic's TOIF, which serves to merge analysis tool results and represent it in a common format. Rather than presenting the end user with one monolithic software system that covers multiple stages of the SwA process, the capabilities offered by these groups allow the integration and manipulation of results from different analyzers.

The value of SWAMP's open and evolving framework reaches far beyond software developers. It will help tool developers, educators, researchers and software consumers in their ongoing effort to improve the software they develop and deploy.

To meet the diverse and ever changing needs and expectations of the different groups that compose the SwA community, the framework will have to offer the following key elements:

An environment where new tools can be added easily and efficiently: Tool developers and researchers should be able to bring their tools into the framework with no more difficulty than bringing the tool up on their own desktop. This means not only having simply uploading procedures, but also being able to work interactively to address porting issues.

Ease of bringing tools into the framework also means that once a tool is available, the operation of running it against a software package should be fully automated. Such automation requires that the framework provide the glue to run the tool against software packages with complex and even non-standard build procedures.

An environment where new software packages can be added easily: As with the case for tools, software package developers should be able to bring their software into the framework with no more difficulty than bringing the package up on their own desktop. Again, interactive access is critical to keep this process simple and familiar. Once a package is successfully built, the effort of the package developer should be done. The framework must provide the glue that automates running selected tools against the package, assessing the package exactly as it would be built, and handling complex directory structures, separate compilation, whole program analysis, and builds that produce multiple executables.

Support for tools that integrate and interpret the output of SwA assurance tools: The step of automating importing tools and software packages, and running the tools against the packages, are only the first steps in the workflow. While running against multiple SwA tools provides a rich source of assessment information, this information must be unified, labeled and presented to the user in a way that allows

them to understand it. The SwA framework must provide open access for tools that fill all or parts of this space.

An open framework with access to software products and results at all levels: An SwA framework will include analysis products from each step of the workflow. These products include the raw results from SwA tools, normalized raw results in a uniform format, merged and interpreted and labeled results, annotated results that include feedback from the programmer or higher level tools. Tool developers and programmers must have the opportunity to access any of these results and share these results, while not being forced to depend on any of them. Common data representations are key to allow the choice of multiple tools at any stage of the SwA process, and to allow independently developed tools to interact with each other.

A foundation for understanding the process of software assessment: The body of data that will be created by an active and productive SwA framework provides a life history of the software development and assurance process. As such, this data offers raw materials for study to the researcher in software engineering, software assurance, risk management, and software business processes. For example, a researcher might be studying the productivity of a software assessment method or the effectiveness of various tools and techniques. Data can be provided to researchers in both raw and anonymized forms.

This is the first in a collection of short whitepapers that we plan to present to the SwA community. Our goal in writing these papers is to introduce the principals and capabilities of the SWAMP and to engage the community in a dialogue. We welcome comments and recommendations that will help us expend the reach and extend the impact of the SWAMP. We plan to start with papers that will discuss how the SWAMP will address each of the key framework elements that were identified in this paper.