Introduction to New Directions in Software Assurance Minitrack

Dan Port
University of Hawaii
dport@hawaii.edu

Jairus Hihn
Joel Wilf
Jet Propulsion Laboratory
California Institute of Technology
jhihn@jpl.nasa.gov
wilf@jpl.nasa.gov

Many companies are struggling with how to assure their software products and processes in an effective manner. Traditional software assurance is often required on large software intensive projects but is typically kept to a minimum level of support because of a perceived lack of value. This perceived lack of value is increasing in spite of software becoming an increasing source of risk not only for large aerospace and defense projects but also in the automotive industry whose most sophisticated vehicles have well over a million lines of code. Part of the issue is that current practices of software assurance are being driven by tradition (e.g. “do what has been done before”) or institutional requirements (e.g. “do what is needed to satisfy policy or regulations), which are increasingly inadequate in providing effective software assurance activities and definitely inadequate in rationalizing its value added to a project.

As part of the focus on software process improvement, various new approaches for improving the quality and acceptance of software assurance that do not rely on these drivers are being considered. At the same time there are other areas where assurance activities are being applied but as performed by the engineering organizations as part of the way they do their work and not being performed by an independent assurance organization.

This minitrack has four papers looking at different ways to apply and improve how software assurance activities are conducted. The first two papers look at how to improve software assurance as performed by the software assurance organizations and the last two look at how to apply software assurance methods outside the software assurance organizations.

1. The first paper looks at the state of practice and state of the art tool usage in the assurance world. It evaluates the current tools in use (e.g. JIRA for tracking risks, static code analyzers and CASRE for analyzing reliability) and identifies major areas where little or no tool support exists. With respect to state of art tools the paper reports on a research task on the use of text mining methods to analyze requirements, risks and defects.

2. The second paper looks at the use visualization techniques to better analyze large software assurance data sets. While visualization techniques, such as Treemap, have long been applied in software engineering, the application to software process and product assurance metrics and findings is relatively novel.

3. The third paper documents an extensive set of cost estimation best practices and then explores whether applying some of the methods used in software assurance might improve the quality of software cost estimates. This paper especially focuses on issues associated with model calibration, estimate review, and the development of a comprehensive basis of estimate.

4. The last paper looks at how to manage, improve and assure large data processing systems by an innovative use of workflow management systems. The authors present an architecture that provides functionality to produce both correct output as well as fast re-execution by leveraging the provenance of data to propagate changes along the implicit dependency graph. By tracking each data object and process, they create a provenance graph and assurance attributes that can help us verify data trustworthiness dynamically, facilitating real-time awareness of each object’s security, re-productibility and correctness in computation.