Integrating security testing into continuous integration workflows

Automated testing has become a normal part of developer life in the world of agile development processes. However, it is usually only focused on unit testing or behavior testing—either the technical or the functional realm of the application. What about security testing and other non-functional aspects of the application, like performance testing?

Especially if you work in strongly regulated environments like government, healthcare and finance, you’ll have to continuously demonstrate that guidelines are being followed when it comes to aspects like input validation, auditability and traceability.

However, automated testing isn’t only for introducing checks and balances after development ends in a commit and a successful build. It is also a very valuable strategy to utilize automated testing tools in the definition phase of a project.

Especially since the rise of Behavior driven development strategy, test tooling has become able to work with native/human readable language scripts, which can both serve as user story in an agile development process, as input for the developer during development, and then as final check during the performance of the tests.

This paradigm change has been implemented by OpenNovations in several organizations ranging from commercial entities in highly regulated pharmaceutical environments and governmental agencies, as well as shared in sessions for DARQA and other network organizations.

This poster outlines number of the practical lessons learned while integrating automated security testing into the regular development strategies, we will look into some open source tools to deploy for security testing and give some practical advise on how to integrate these with your Continuous Integration setup:

- static code analysis for coding style and technical debt
- security testing tools
- performance and UI testing tools