Software Test Automation

Oliver Tuschling, 2008

Automated software testing can significantly reduce the effort required for adequate testing, or significantly increase the testing which can be done in limited time. Tests can be run in minutes that would take hours to be run manually. Automation can enable our organization to produce better quality software more quickly than would have been possible by manual testing alone.
One of the main benefits of software automation is that tests are repeatable, using exactly the same inputs in the same sequence time and again, something that cannot be guaranteed with manual testing.

Usually, test automation is synonymous with capture/playback tools. However, the traditional capture/playback method of automation so prevalent in vendor demonstrations is not adequate to address the needs of complex, multi-environment systems. Both manual and automated scripts are required to provide the robustness, maintainability, and reusability so crucial to long-term automation success.

Benefits of test automation:

- Run existing batteries of automated test cases on a new version of our product (regression tests)
- Run more automated tests more often.
- Perform tests that may be impossible to be done manually. For example, simulating 50 people doing actions in a specific product at the same time may be simulated with, for example 5 virtual images running with automatic tests distributed at them.
- Consistency and repeatability of tests. Same tests can be executed on different hardware configurations, in different operating systems or against different databases.
- Increased confidence on our products quality. Knowing that an extensive set of automated tests has run successfully, there can be a greater confidence that the product has more quality and there won't be any unpleasant surprises when the system is released.
- Develop and run low level tests (also unit tests) just after the product is compiled and before it is delivered to the software quality department.

Talking about software test automation makes necessary to talk about software test automation tools. There are several tools that can be used to automate software tests. There are GUI test tools like SilkTest, Winrunner, WhiteApi (Open Source) that are used for testing the application user interface, also low level test tools like nUnit, Visual Studio Test Infrastructure which allow testing of specific low level components. An organization can also develop powerful internal test tools.

Automating tests can be also creating repeatable processes that may be run manually but following a script or a process that contains some prepared parts of automation. A good example for this is to launch some input data to an application using a data simulator, and launching a battery of prepared database queries that validates data is properly stored in database.

For companies with a complex application suite that contain lots of areas in there it is necessary to go to different phases during an specific project (Qualification Test, Acceptance Test, New features test, Regression Test, etc…) as well as different types of test (functional, performance-capacity, components…). For these companies, it is important to have an automation test team investment that may be focused on Automating partially or totally as much part of software tests as possible, based on a prioritized list of areas and strategies already defined and analyzed.

A software automation test team should be focused on:

- Fully or partial automate new areas, according to the SQA strategy and based on prioritized lists.
- Maintain and adapt existing automated suites to new versions of the products.
- Create new automated tests for existing areas that are candidate to be automated.
- Manage outsourced automation tasks (if existing)
- Create processes to support people on optimizing new test cases.
- Create useful custom tools to help on testing.
- Review old existing test cases to see how they can be optimized and prepared to be automated in the future.

A software automation test team should have at least 2 main roles clearly defined:

Automation Test Lead: Designs and develops automation architecture, assists in design and development of test automation across projects. Has an excellent knowledge of test automation tool and client's practices and procedures, development skills.

Automation Test Engineer: Designs and develops automation scripts. These may be developed from manual scenarios. Good knowledge of tools and testing. Design and configure test environment. Update automation software. Has excellent knowledge of test environments, tools and practices and procedures.