

Automated System Testing with Selenium2

Demonstrate how to use Selenium to test a modern interactive JavaScript driven application, and some of the problems involved with automated GUI testing. Demonstrate your solution to the exercise “Automated System Testing with Selenium2”

Discuss Pros and Cons with manual versus automated tests

- In manual testing, test cases are executed manually without any support from tools or scripts. But with automated testing, test cases are executed with the assistance of tools, scripts and software.
- The type of testing (manual or automated) depends on project requirements, budget, timeline etc.

Manual testing is best in the following scenarios:

- **Exploratory Testing** – requires tester’s knowledge, experience, analytical skills and creativity. The test is characterized by poorly written specification documentation.
- **Usability Testing** – you need to measure how user-friendly, efficient or convenient the software is for the end users. Human observation is the most important factor here.
- **Ad-hoc Testing** – there is no specific approach, its unplanned method of testing.

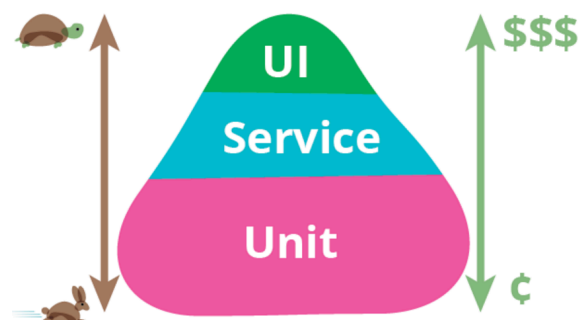
Automated testing is best in the following scenarios

- **Regression Testing** – here, automated testing is suitable because of frequent code changes and the ability to run the regressions in a timely manner.
- **Load Testing** – Automated testing is the best way to complete load testing efficiently.
- **Repeated Execution** – the repeated execution of a task is best automated.
- **Performance Testing** – testing which requires simulation of many concurrent users.

Explain about the Test pyramid and whether this exercise supported the ideas in the “pyramid”

The point of the Testing Pyramid is that you should have many more low-level unit tests than high level end-to-end tests running through a GUI since:

- Cost of tests goes up, as you go up in the pyramid
- Coverage is higher closer to the bottom
- UI tests are typically slow
- UI tests requires larger setup



AUTOMATED SYSTEM TESTING

Discuss some of problems with automated GUI tests and what makes such tests “vulnerable”

GUI tests executed to stimulate user interacting with the system.

- GUI automated tests can be very breakable, due to UI changes even though functionality hasn't changed. Slow feedback to the team.
- Execution is slow, because you have to wait for the system to launch and connect.
- You are dependent on 3rd party tools for GUI testing.

Demonstrate details in how to create a Selenium Test using the code for the exercise

Selenium is a free (open source) automated testing tool for web applications across different browsers and platforms. We have worked with Selenium 2, which is a combination of WebDriver and Selenium RC.

Explain shortly about the DOM, and how you have read/manipulate DOM-elements in your test

- With the HTML DOM, JavaScript can access and change all the elements of an HTML document.
- When a web page is loaded, the browser creates a DOM of the page.
- DOM can be manipulated programmatically, with Selenium tests.

Explain how (and why it was necessary) you have solved “waiting” problems in your test

It is important to solve the waiting problems when working with selenium tests, because if you set the timer to 10 seconds, and the test fails, you have just wasted time on waiting for the test to fail.