



# Test Automation & Advanced Selenium

**Balaji Dinakaran**

[balajidinakaran1@gmail.com](mailto:balajidinakaran1@gmail.com)

<https://www.linkedin.com/in/balaji-dinakaran/>

## I: Introduction to Automation

# Lesson Objectives

- Automation Vs. Manual
- What is Automation
- What is Test Automation
- Why And When
- Example Of Test Automation



# I.I Introduction to Automation

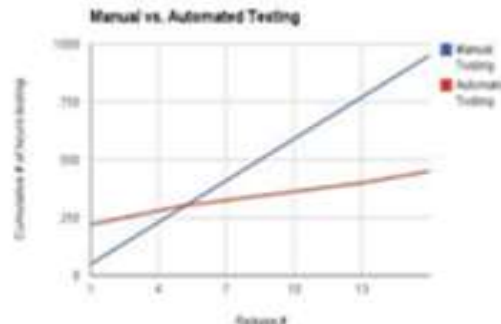
## Automation Vs. Manual

### Manual Testing

- Testing is time consuming and tedious
- Delay the ability in thoroughly testing an application
- Critical bugs escape undetected
- What happens when multiple platforms involved

### Automated Testing

- Higher efficiency
- Higher product quality
- Easy to focus on all possible workflows
- Delivers: Reusability, Consistency and Productivity



# What is Automation

“The first rule of any technology used in a business is that automation applied to an efficient operation will magnify the efficiency. The second is automation applied to an inefficient operation will magnify the inefficiency.” - Bill Gates





# What is Test Automation

- The method which /that takes automation tool's support to execute the test cases is known as Automation Testing.
- It is a method which
  - Uses automation tools to run tests that repeat predefined actions
  - Matches the developed program's probable and real results
- If the project prospects and results align, your project is behaving as it should, and you are likely bug free. If the two don't align, still, here is a problem that requires to be addressed. You'll have to take a look at your code, alter it, and continue to run tests until the actual and expected outcomes again.



# Test Automation and its Benefits:

## **Why automate Testing?**

➤ In today's fast moving world, it is a challenge for any company to continuously maintain and improve the quality and efficiency of software systems development. In many software projects, testing is neglected because of time or cost constraints. This leads to a lack of product quality, followed by customer dissatisfaction and ultimately to increased overall quality costs.

## **The main reasons for these added costs are primarily:**

- poor test strategy
- underestimated effort of test case generation
- delay in testing
- subsequent test maintenance



# Test Automation and its Benefits:

- Test automation can improve the development process of a software product in many cases. The automation of tests is initially associated with increased effort, but the related benefits will quickly pay off.
- Automated tests can run fast and frequently, which is cost-effective for software products with a long maintenance life. When testing in an agile
- environment, the ability to quickly react to ever-changing software systems and requirements is necessary. New test cases are generated continuously and can be added to existing automation in parallel to the development of the software itself.
- In both manual and automated testing environments test cases need to be modified for extended periods of time as the software project progresses. It is important to be aware that complete coverage of all tests using test automation is unrealistic. When deciding what tests to automate first, their value vs. the effort to create them needs to be considered. Test cases with high value and low effort should be automated first. Subsequently test cases with frequent use, changes, and past errors; as well as test cases with low to moderate effort in setting up the test environment and developing the automation project are best suited for automation.





# Optimization of Speed, Efficiency, Quality and the Decrease of Costs:

➤ The main goal in software development processes is a timely release. Automated tests run fast and frequently, due to reused modules within different tests. Automated regression tests which ensure the continuous system stability and functionality after changes to the software were made lead to shorter development cycles combined with better quality software and thus the benefits of automated testing quickly outgain the initial costs.



# Advance a Tester's Motivation and Efficiency:

➤ Manual testing can be mundane, error-prone and therefore become exasperating. Test automation alleviates testers' frustrations and allows the test execution without user interaction while guaranteeing repeatability and accuracy. Instead testers can now concentrate on more difficult test scenarios.



# Increase of Test Coverage:

- Sufficient test coverage of software projects is often achieved only with great effort. Frequent repetition of the same or similar test cases is laborious and time consuming to perform manually.

# Principles of Testing

- (i) All the test should meet the customer requirements
- (ii) To make our software testing should be performed by third party
- (iii) Exhaustive testing is not possible. As we need the optimal amount of testing based on the risk assessment of the application.
- (iv) All the test to be conducted should be planned before implementing it
- (v) It follows pareto rule(80/20 rule) which states that 80% of errors comes from 20% of program components.
- (vi) Start testing with small parts and extend it to large parts.

# Types of Testing

- Unit Testing
- Integration Testing
- System Testing
- Sanity Testing
- Smoke Testing
- Interface Testing
- Regression Testing
- Beta/Acceptance or Alpha Testing

# Unit Testing

It focuses on smallest unit of software design. In this we test an individual unit or group of inter related units. It is often done by programmer by using sample input and observing its corresponding outputs.

Example:

- a) In a program we are checking if loop, method or function is working fine
- b) Misunderstood or incorrect, arithmetic precedence.
- c) Incorrect initialization

# Integration Testing

Testing of all integrated modules to verify the combined functionality after integration is termed as Integration Testing.

Modules are typically code modules, individual applications, client and server applications on a network, etc. This type of testing is especially relevant to client/server and distributed systems.

**Integration Test aims to test how well several units interact with each other.**

Example

(a) Black Box testing:- It is used for validation.

In this we ignore internal working mechanism and focuses on what is the output?.

(b) White Box testing:- It is used for verification.

In this we focus on internal mechanism i.e.  
how the output is achieved?



# Regression Testing

Testing an application as a whole for the modification in any module or functionality is termed as Regression Testing. It is difficult to cover all the system in Regression Testing, so typically Automation Testing Tools are used for these types of testing.

## Example

In school record suppose we have module staff, students and finance combining these modules and checking if on integration these module works fine completely is regression testing



# Smoke Testing

- Whenever a new build is provided by the development team then the Software Testing team validates the build and ensures that no major issue exists.
- The testing team ensures that the build is stable and a detailed level of testing is carried out further. Smoke Testing checks that no show stopper defect exists in the build which will prevent the testing team to test the application in detail.
- If testers find that the major critical functionality is broken down at the initial stage itself then testing team can reject the build and inform accordingly to the development team. Smoke Testing is carried out to a detailed level of any Functional or Regression Testing.

# Sanity Testing

- Sanity Testing is done to determine if a new software version is performing well enough to accept it for a major testing effort or not. If an application is crashing for the initial use then the system is not stable enough for further testing. Hence a build or an application is assigned to fix it.



# Smoke vs Sanity Testing

- **Smoke testing** means to verify (basic) that the implementations done in a build are working fine.
- **Sanity testing** means to verify the newly added functionalities, bugs etc. are working fine



# Alpha Testing

This is a type of validation testing. It is a type of acceptance testing which is done before the product is released to customers. It is typically done by QA people.

Example:

When software testing is performed internally within the organization.



# Beta Testing

The beta test is conducted at one or more customer sites by the end-user of the software. This version is released for the limited number of users for testing in real time environment

Example:

When software testing is performed for the limited number of people

# System Testing

In this software is tested such that it works fine for different operating system. It is covered under the black box testing technique. In this we just focus on required input and output without focusing on internal working.

In this we have security testing, recovery testing , stress testing and performance testing

Example:

This include functional as well as non functional testing

# Stress Testing

In this we give unfavorable conditions to the system and check how they perform in those conditions.

Example:

- (a) Test cases that require maximum memory or other resources are executed
- (b) Test cases that may cause thrashing in a virtual operating system
- (c) Test cases that may cause excessive disk requirement



# Performance Testing

It is designed to test the run-time performance of software within the context of an integrated system. It is used to test speed and effectiveness of program.

Example:

Checking number of processor cycles.

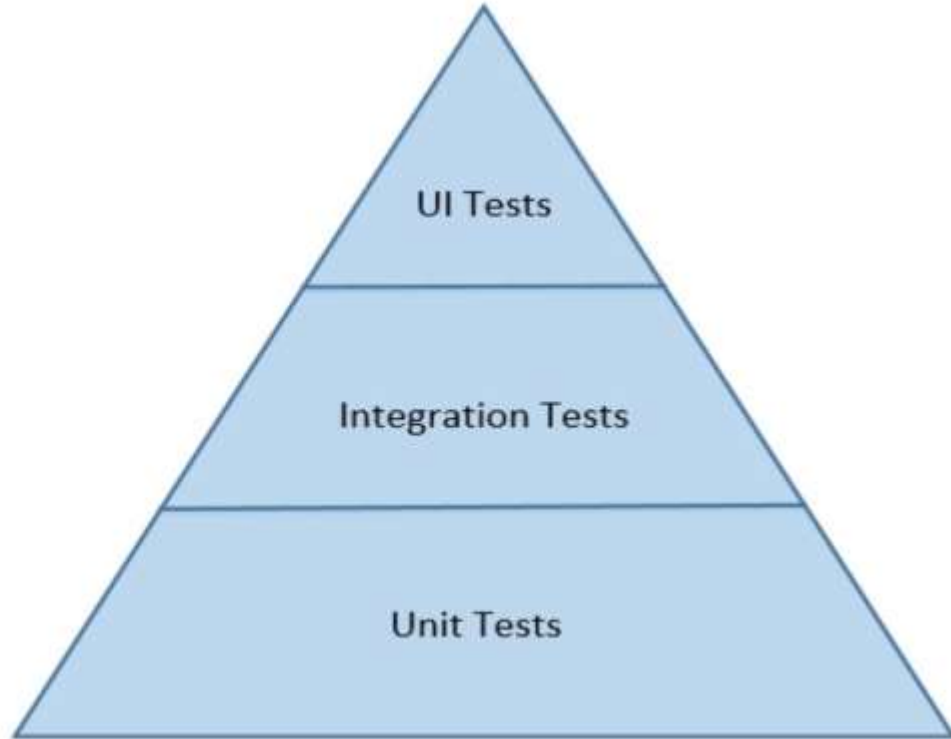


# Test Pyramid

High Integration  
Slow Execution  
Lesser in Number



High Isolation  
Fast Execution  
Higher in Number



# Characteristics of a good unit test

- **Fast.** It is not uncommon for mature projects to have thousands of unit tests. Unit tests should take very little time to run. Milliseconds.
- **Isolated.** Unit tests are standalone, can be run in isolation, and have no dependencies on any outside factors such as a file system or database.
- **Repeatable.** Running a unit test should be consistent with its results, that is, it always returns the same result if you do not change anything in between runs.
- **Self-Checking.** The test should be able to automatically detect if it passed or failed without any human interaction.
- **Timely.** A unit test should not take a disproportionately long time to write compared to the code being tested. If you find testing the code taking a large amount of time compared to writing the code, consider a design that is more testable.



## Some examples are:

- Regression test after debugging or further development of software
- Testing of software on different platforms or with different configurations
- Data-driven testing (creation of tests using the same actions but with many different inputs)
- Test automation allows performing different types of testing efficiently and effectively.

# When & Why

- Tests can run fast and frequently
  - Cost-effective for software products with a long maintenance life
- Useful in agile environment
  - Robust test automation projects balanced for value and effort
- optimization of Efficiency & Quality
  - Quick return on investment(ROI) of test automation
- Advanced a tester's motivation and efficiency
  - More efficient assignments of QA Tasks
- Increase of Test Coverage
  - Different types of testing to increase test coverage

# Example of Test Automation

## ➤ Test sample for invalid Date

```
@Test
public void Add_User_With_Invalid_Data()
{
    /*Pre Conditions
    * 1. Check login
    * 2. Check logged in user has access to manage users
    * */

    /* Adding a user
    * 1. Navigate to Manage Users Grid
    * 2. Check for "Add Users" button, if exists,
    * 2.1 Click on Add user button in Manage users Grid
    * 3. Check for the Fields and Enter required fields if exists
    * 3. Click on Submit button
    * 4. Error message should be displayed.
    * 5. Compare for the Error message based on the input provided
    * */
}
```

## In Above example “Add Users With Valid Data”

➤ To execute the test case, we need to login to the application. We also need to check if the user is already logged in or not. And the other we need to check is if the logged in person is having access to “Add Users” or Not. If the above Two Conditions are passed then we should execute the rest Else we should return the test as failed.

# Summary

## In this lesson, you have learnt

- Testing is an extremely creative & intellectually challenging task
- Manual testing is performed by a human sitting in front of a computer carefully executing the test steps
- Automation testing means using an automation tool to execute your test cases suite
- Goal of automation is to reduce number of test cases to be run manually and not eliminate manual testing all together.



# Review Question

## Question 1

- Why would you want to automate a test? List it to:
- Increase test coverage?
- Improve quality?
- Save time for exploratory testing?



## Question 2: True/false

- Automation testing uses automation tools to run tests that
- Repeat predefined actions.

## Question 3: Fill in the Blanks

- Automation testing delivers \_\_, consistency and productivity.





# Test Automation & Advanced Selenium

## Lesson2:Introduction to selenium

# Lesson Objectives

- Introduction Selenium
- What it is and what it is not
- Landscape and usage
  - Selenium IDE
  - Selenium remote control(selenium 1.0)
  - Selenium web driver(selenium 2.0)
  - Selenium grid



# Introduction to Selenium

- Invented by Jason R. Huggins
- Free (open source) automated testing suite for web applications across different browsers and platforms
- **Selenium has a suite of tools which consists of the following:**
  - selenium IDE
  - selenium RC(remote control), also known as selenium 1
  - selenium web driver or selenium 2
  - selenium Grid
- Test cases in selenium can be written in many popular programming languages supported by selenium like java, C#, Ruby, Python etc.

# Selenium: What it is?

- Selenium automates browsers. That's it!
- Selenium is a portable testing API for web application
- Web application : Pure HTML & JavaScript(No Flash, ActiveX, Silver light etc.)
- Primary it is for automating web applications for testing purposes, but is certainly not limited to just that.
- It is open source and freeware
- Support multiple browsers & OS.
  - Browsers: firefox, chrome, IE, safari, opera, phantomJS
  - OS: windows, MAC, Linux, Android, IOS
- In process headless execution.

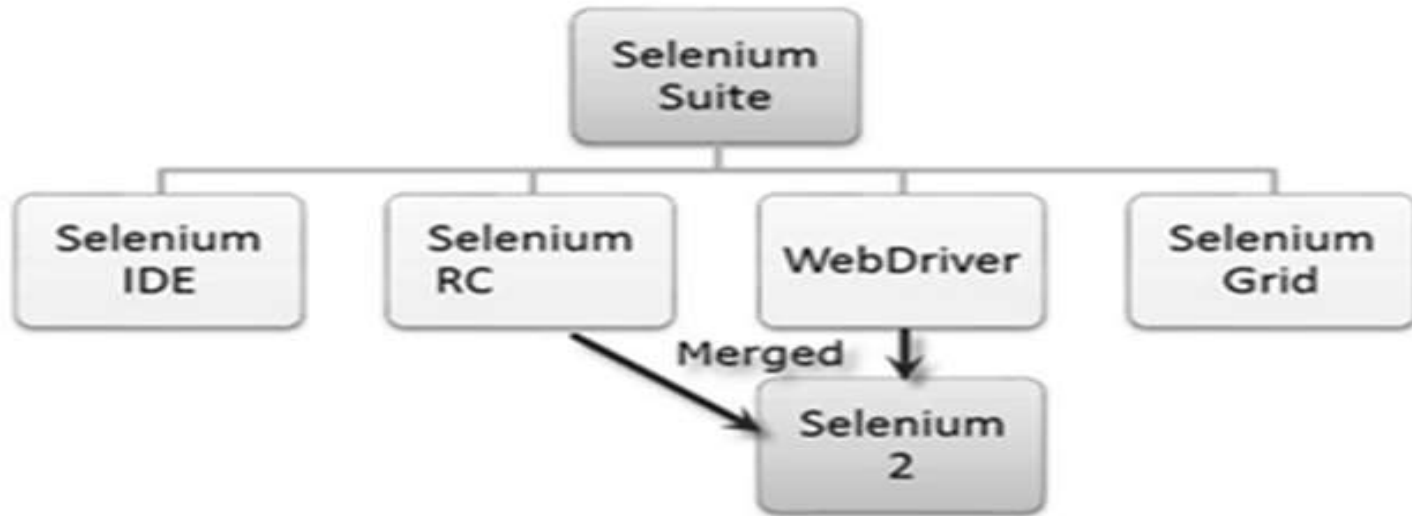


# Selenium: What it is NOT?

- Selenium is NOT a test tool
- It is NOT a framework
- It is NOT a installable software
- It is NOT a programming language.

# Landscape and Usage

➤ Selenium is not just a single tool but a suite of software's, each catering to different testing needs of an organization. It has four components:

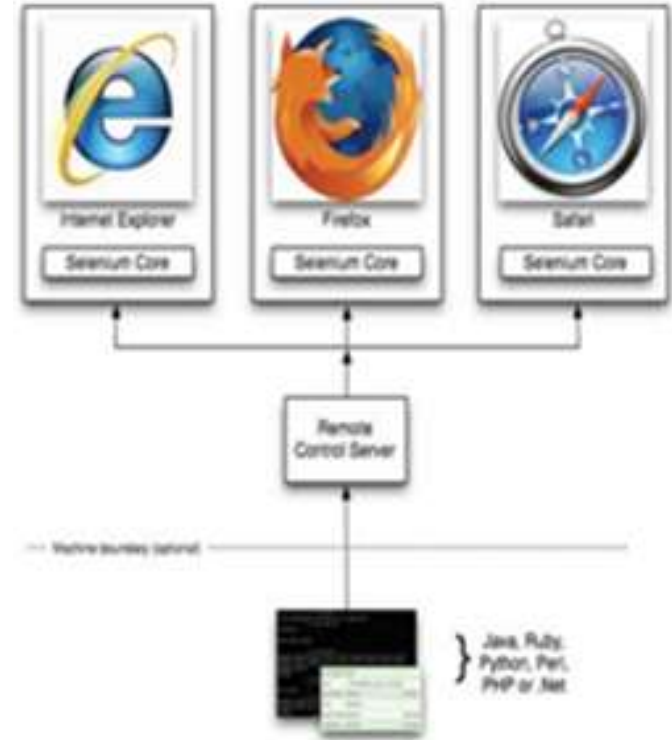


# Overview of Selenium IDE

- Firefox plug in, allows you to record, play back, edit, and debug tests in browser.
- Allows you to record user actions on browser window.
- generate scripts from recorded user actions in most of the popular languages like Java, C#, Perl, Ruby etc. however to run them in an automated testing fashion you need to use selenium web driver.
- Selenium default scripts are html(added JavaScript) and that is the scripts we are going to use it in selenium IDE
- The reason for availability of other languages is, user can get the scripts for selenium web driver/RC.
- It also has a context menu (right-click) integrated with the Firefox browser, which allows the user to pick from a list of assertions and verifications for the selected location

# Selenium Remote Control(Selenium 1.0)

- Selenium Remote control (RC) is a test tool that allows you to write automated web application UI tests in any programming language against any HTTP website using any mainstream JavaScript-enabled browser.
- **Selenium RC comes in two parts:**
  - A server which automatically launches and kills browsers, and acts as a HTTP proxy for web requests from them.
  - Client libraries for your favorite computer language.



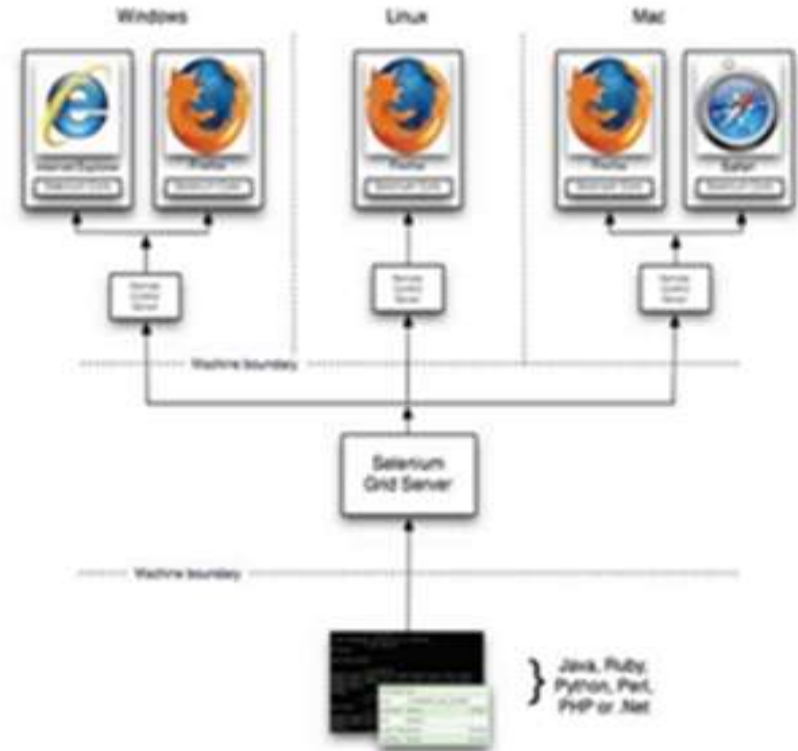


# Selenium Web Driver(Selenium 2.0)

- Selenium Web Driver proves itself to be better than both selenium IDE and selenium RC in many aspects.
- It implements a more modern and stable approach in automating the browser's actions.
- Web Driver, unlike selenium RC, does not rely on JavaScript for automation
- it controls the browser by directly communicating to it.

# Selenium Grid

- Runs multiple tests on different machines against different browsers and operating systems in parallel
- Supports distributed test execution.
- Run tests in a distributed test execution environment



# Introduction to Framework

- **Data Driven Test Framework**

- In data driven framework all of our test data is generated from some external files like Excel, CSV, XML, JSON or some database table

- **Keyword Driven Test Framework:**

- In keyword driven test framework, all the operations and instructions are written in some external file like Excel worksheet

# TestNG Framework

## What is TestNG?

- Automated testing framework
- NG = Next Generation
- Similar to JUnit (especially JUnit 4)
- Not a JUnit extension (but inspired by JUnit)
- Designed to be better than JUnit, especially when testing integrated classes
- Created by Dr. Cédric Beust (of Google)
- Open source (<http://testng.org>)

# Page Object Model Patten

- **Page Object Model** is a design pattern to create **Object Repository** for web UI elements.
- Under this model, for each web page in the application, there should be corresponding page class.
- This Page class will find the WebElements of that web page and also contains Page methods which perform operations on those WebElements.
- Name of these methods should be given as per the task they are performing, i.e., if a loader is waiting for the payment gateway to appear, POM method name can be **waitForPaymentScreenDisplay()**.

# Advantages of POM

- Page Object Pattern says operations and flows in the UI should be separated from verification. This concept makes our code cleaner and easy to understand.
- The Second benefit is the object repository is independent of test cases, so we can use the same object repository for a different purpose with different tools. For example, we can integrate POM with TestNG/JUnit for functional Testing and at the same time with JBehave/Cucumber for acceptance testing.
- Code becomes less and optimized because of the reusable page methods in the POM classes.
- Methods get more realistic names which can be easily mapped with the operation happening in UI. i.e. if after clicking on the button we land on the home page, the method name will be like 'gotoHomePage()'.

# Summary

## In this lesson, you have learnt

- Selenium is a free (open source) automated testing suite for web applications across different browsers and platforms
- Selenium is a collection of different tools
- Selenium IDE, a Firefox add-on that you can only use in creating relatively simple test cases and test suites
- Selenium RC allows users to use programming languages in creating complex tests.
- Web Driver, allows test scripts to communicate directly to the browser, thereby controlling it from the OS level.
- Selenium Grid is used with selenium RC to execute parallel tests across different browsers and operating systems.

