KAUSHALYA TECHNICAL TRAINING AND CONSULTANCY SERVICES

Raghu Prasad

9845547471

www.kaushalya.tech

FB : https://www.facebook.com/raghuprasadkonandur/

Skill Up-Skill Re-Skill

Excellence in action

Introduction

- Raghu Prasad – BE, MS
- Total of 23 years of experience
- 7 years as a lecturer in an Engineering College
- 16 Years into IT
- Worked with companies like CISCO,CSC,ICICI,First Apex – NTT Data
- Currently into Corporate training and consultancy
- Worked with corporates and public sector
- **Service Offerings –** In person/On-line/Corporate/Academic Institutes
- **Technologies –** Java,Python,Web technologies,Java Script technologies (MEAN stack),IOT,Test Automation,Machine Learning,Artificial Intelligence,ERP
- **Customers –** L & T,NextGen,Incarnus,BGS-IT,Sindhi College,Aspire Technologies

Topics

- Introduction to Test Automation
- Introduction to Selenium
- Introduction to Java and deep dive – Hands-on
- Introduction to HTML/CSS/Java Script – Hands-on
- Deep dive into selenium web driver – Hands-on
- Introduction to Apache Jmeter and deep dive – Hands-on

What is Test Automation ..

- **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 case suite.
- In software testing, **test automation** is the use of special software (separate from the software being tested) to control the execution of tests and the comparison of actual outcomes with predicted outcomes.
- **Test automation** can automate some repetitive but necessary tasks in a formalized testing process already in place, or perform additional testing that would be difficult to do manually. Test automation is critical for continuous delivery and continuous testing.

What is Test Automation ?

- Test Automation demands considerable investments of money and resources.
- Using a test automation tool it's possible to record this test suite and re-play it as required.
- Once the test suite is automated, no human intervention is required .This improved ROI of Test Automation.
- Goal of Automation is to reduce number of test cases to be run manually and not eliminate Manual Testing all together.

What is Test Automation ?

TESTING PROCESS IN MANUAL AND AUTOMATION TESTING

Manual Testing:

Automation Testing:

Why Automation Testing ?

- Automated software testing is important due to following reasons:

✓Manual Testing of all work flows, all fields , all negative scenarios is time and cost consuming

✓It is difficult to test for multi lingual sites manually

✓Automation does not require Human intervention. You can run automated test unattended (overnight)

✓Automation increases speed of test execution

✓Automation helps increase Test Coverage

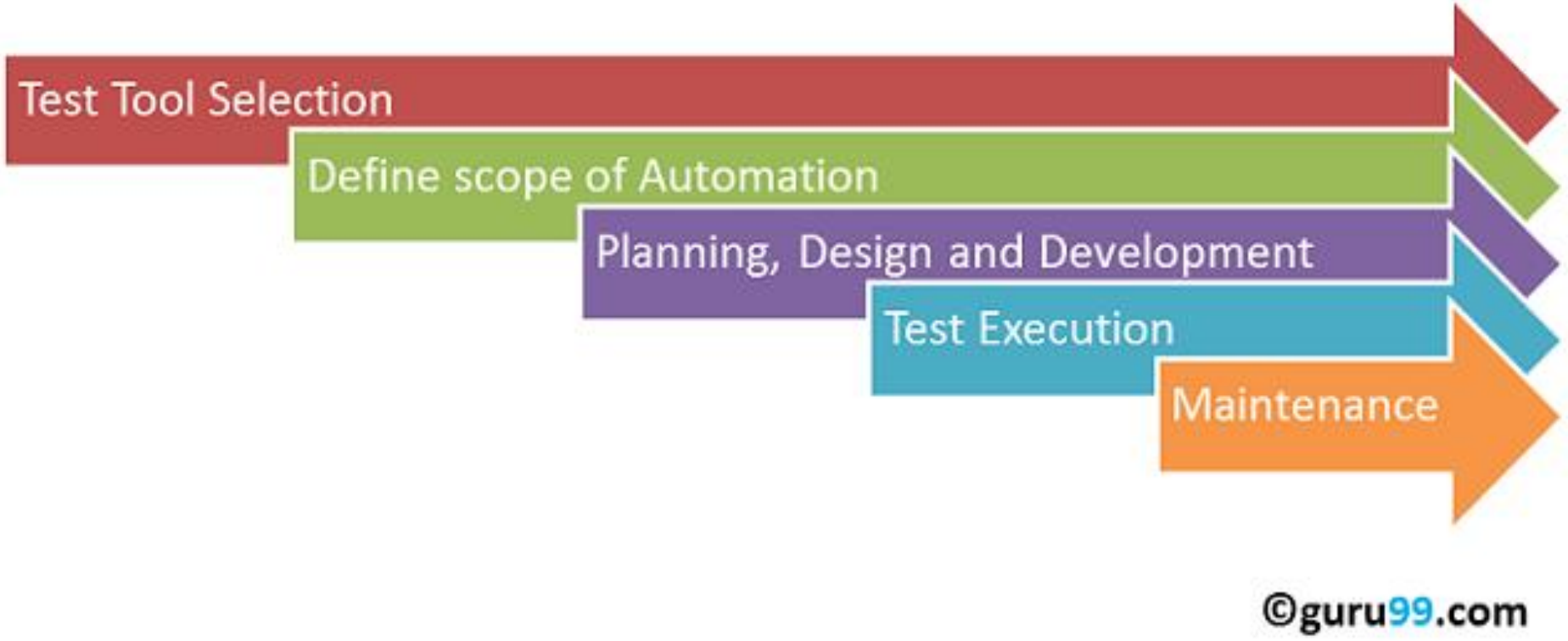
✓Manual Testing can become boring and hence error prone.

Which test cases to Automate

- Test cases to be automated can be selected using the following criterion to increase the automation ROI
  - ✓High Risk - Business Critical test cases
  - ✓Test cases that are executed repeatedly
  - ✓Test Cases that are very tedious or difficult to perform manually
  - ✓Test Cases which are time consuming
- The following category of test cases are not suitable for automation:
  - ✓Test Cases that are newly designed and not executed manually at least once
  - ✓Test Cases for which the requirements are changing frequently
  - ✓Test cases which are executed on ad-hoc basis.

Automation Testing process

- Following steps are followed in an Automation Process



Test tool selection

- Test Tool selection largely depends on the technology the Application Under Test is built on.
- For instance QTP does not support Informatica. So QTP cannot be used for testing Informatica applications.
- It's a good idea to conduct Proof of Concept of Tool on automation

Define scope of automation

- Scope of automation is the area of your Application Under Test which will be automated. Following points help to determine the scope:
  - ✓Feature that are important for the business
  - ✓Scenarios which have **large amount of data**
  - ✓**Common functionalities** across applications
  - ✓Technical feasibility
  - ✓Extent to which business components are reused
  - ✓**Complexity** of test cases
  - ✓Ability to use the same test cases for cross browser testing

Planning design and development

- During this phase you create Automation strategy & plan, which contains following details-
  - ✓Automation tools selected
  - ✓Framework design and its features
  - ✓In-Scope and Out-of-scope items of automation
  - ✓Automation test bed preparation
  - ✓Schedule and Timeline of scripting and execution
  - ✓Deliverables of Automation Testing

Test execution

- Automation Scripts are executed during this phase. The scripts need input test data before there are set to run. Once executed they provide detailed test reports.
- Execution can be performed using the automation tool directly or through the Test Management tool which will invoke the automation tool.
- Example: Quality center is the Test Management tool which in turn it will invoke QTP for execution of automation scripts. Scripts can be executed in a single machine or a group of machines. The execution can be done during night , to save time.

Maintenance

- As new functionalities are added to the System Under Test with successive cycles, Automation Scripts need to be added, reviewed and maintained for each release cycle.**Maintenance becomes necessary to improve effectiveness of Automation Scripts.**

Framework for automation

- A framework is set of automation guidelines which help in
  - ✓Maintaining consistency of Testing
  - ✓Improves test structuring
  - ✓Minimum usage of code
  - ✓Less Maintenance of code
  - ✓Improve re-usability
  - ✓Non Technical testers can be involved in code
  - ✓Training period of using the tool can be reduced
  - ✓Involves Data wherever appropriate

Framework for automation

- There are four types of framework used in automation software testing:
  1. Data Driven Automation Framework
  2. Keyword Driven Automation Framework
  3. Modular Automation Framework
  4. Hybrid Automation Framework

Automation tool best practices

- To get maximum ROI of automation, observe the following
- Scope of Automation needs to be determined in detail before the start of the project. This sets expectations from Automation right.
- Select the right automation tool: A tool must not be selected based on its popularity but it's fit to the automation requirements.
- Choose appropriate framework
- Scripting Standards- Standards have to be followed while writing the scripts for Automation .Some of them are-
  - Create uniform scripts, comments and indentation of the code
  - Adequate Exception handling - How error is handled on system failure or unexpected behaviour of the application.
  - User defined messages should be coded or standardized for Error Logging for testers to understand.
- Measure metrics- Success of automation cannot be determined by comparing the manual effort with the automation effort but by also capturing the following metrics.
  - Percent of defects found
  - Time required for automation testing for each and every release cycle
  - Minimal Time taken for release
  - Customer satisfaction Index
  - Productivity improvement

Benefit of automation testing

- Following are benefits of automated testing:

Automation tool best practices

- 70% faster than the manual testing
- Wider test coverage of application features
- Reliable in results
- Ensure Consistency
- Saves Time and Cost
- Improves accuracy
- Human Intervention is not required while execution
- Increases Efficiency
- Better speed in executing tests
- Re-usable test scripts
- Test Frequently and thoroughly
- More cycle of execution can be achieved through automation
- Early time to market

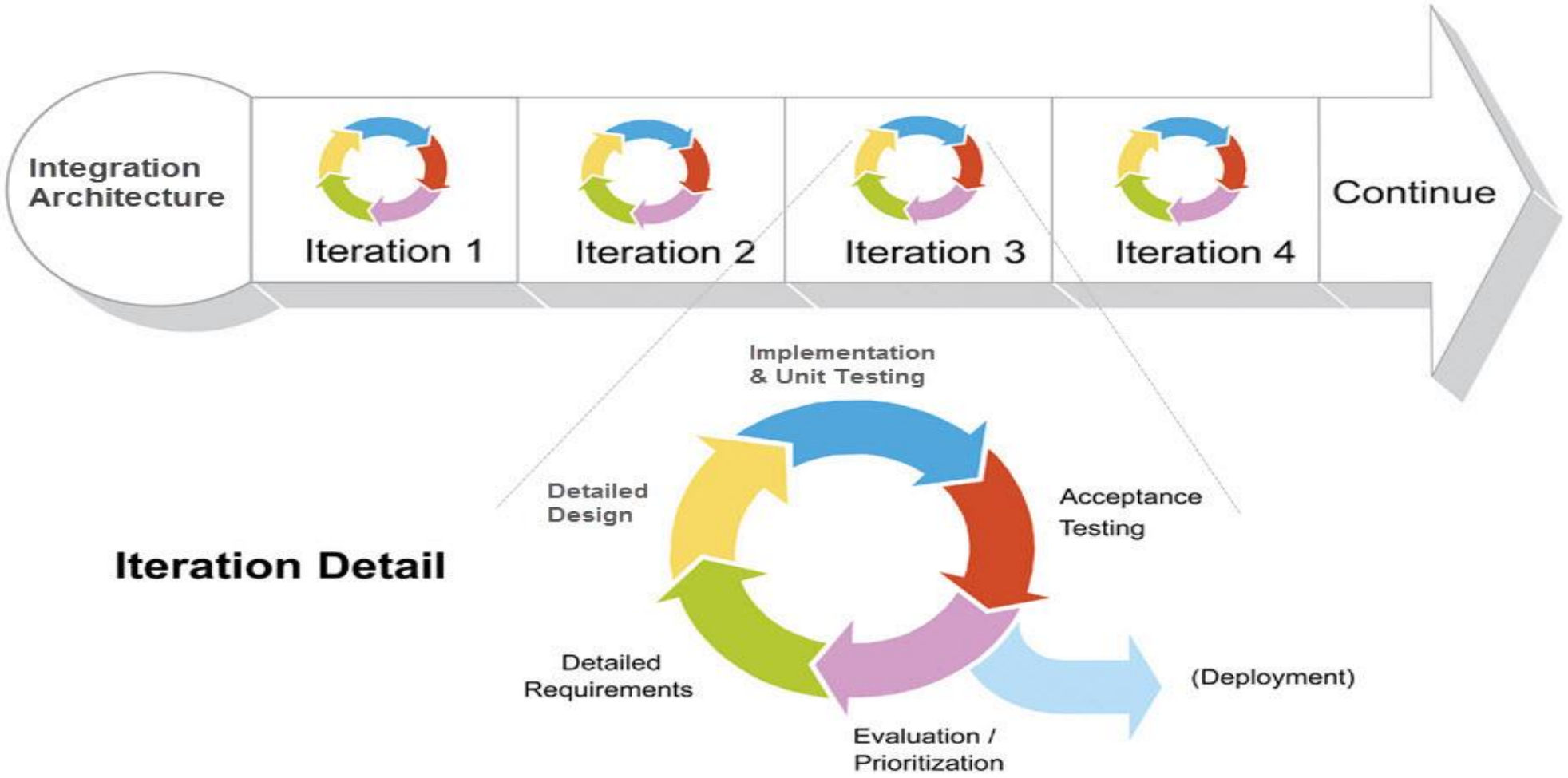
Different types of software testing that can be automated

- Smoke Testing
- Unit Testing
- Integration Testing
- Functional Testing
- Keyword Testing
- Regression Testing
- Data Driven Testing
- Black Box Testing

How to choose automation tool

- **Selecting the right tool can be a tricky task. Following criterion will help you select the best tool for your requirement-**
  - ✓ **Environment Support**
  - ✓ **Ease of use**
  - ✓ **Testing of Database**
  - ✓ **Object identification**
  - ✓ **Image Testing**
  - ✓ **Error Recovery Testing**
  - ✓ **Object Mapping**
  - ✓ **Scripting Language Used**
  - ✓ **Support for various types of test - including functional, test management, mobile, etc...**
  - ✓ **Support for multiple testing frameworks**
  - ✓ **Easy to debug the automation software scripts**
  - ✓ **Ability to recognize objects in any environment**
  - ✓ **Extensive test reports and results**
  - ✓ **Minimize training cost of selected tools**
  - ✓ **Tool selection is one of biggest challenges to be tackled before going for automation. First, Identify the requirements, explore various tools and its capabilities, set the expectation from the tool and go for a Proof Of Concept.**

Agile and automation



Automation testing tools

Top 3 automation tools
1. Selenium

- It is a software testing tool used for Regression Testing. It is an open source testing tool that provides playback and recording facility for Regression Testing. The Selenium IDE only supports Mozilla Firefox web browser.
- It provides the provision to export recorded script in other languages like Java, Ruby, RSpec, Python, C#,etc
- It can be used with frameworks like Junit and TestNG
- It can execute multiple tests at a time
- Autocomplete for Selenium commands that are common
- Walkthrough tests
- Identifies the element using id, name , X-path, etc.
- Store tests as Ruby Script, HTML, and any other format
- It provides an option to assert the title for every page
- It supports selenium user-extensions.js file
- It allows to insert comments in the middle of the script for better understanding and debugging

Top 3 automation tools
2. QTP (Microfocus UFT)

- QTP is widely used for functional and regression testing, it addresses every major software application and environment. To simplify test creation and maintenance, it uses the concept of keyword driven testing. It allows the tester to build test cases directly from the application.
- It is easier to use for non-technical person to adapt to and create working test cases
- It fix defects faster by thoroughly documenting and replicating defects for developer
- Collapse test creation and test documentation at a single site
- Parameterization is easy than WinRunner
- QTP supports .NET development environment
- It has better object identification mechanism
- It can enhance existing QTP scripts without "Application Under Test" being available, by using the ActiveScreen

Top 3 automation tools
3. Rational Functional Tester

- It is an Object-Oriented automated Functional Testing tool that is capable of performing automated functional, regression, data-driven testing and GUI testing. The main features of this tool are
- It supports a wide range of protocols and applications like Java, HTML, .NET, Windows, SAP, Visual basic, etc.
- It can record and replay the actions on demand
- It integrates well with source control management tools such as Rational Clear Case and Rational Team Concert integration
- It allows developers to create keyword associated script so that it can be re-use
- Eclipse Java Developer Toolkit editor facilitates the team to code test scripts in Java with Eclipse
- It supports custom controls through proxy SDK (Java/.Net)
- It supports version control to enable parallel development of test scripts and concurrent usage by geographically distributed team
-

Automation tools comparison

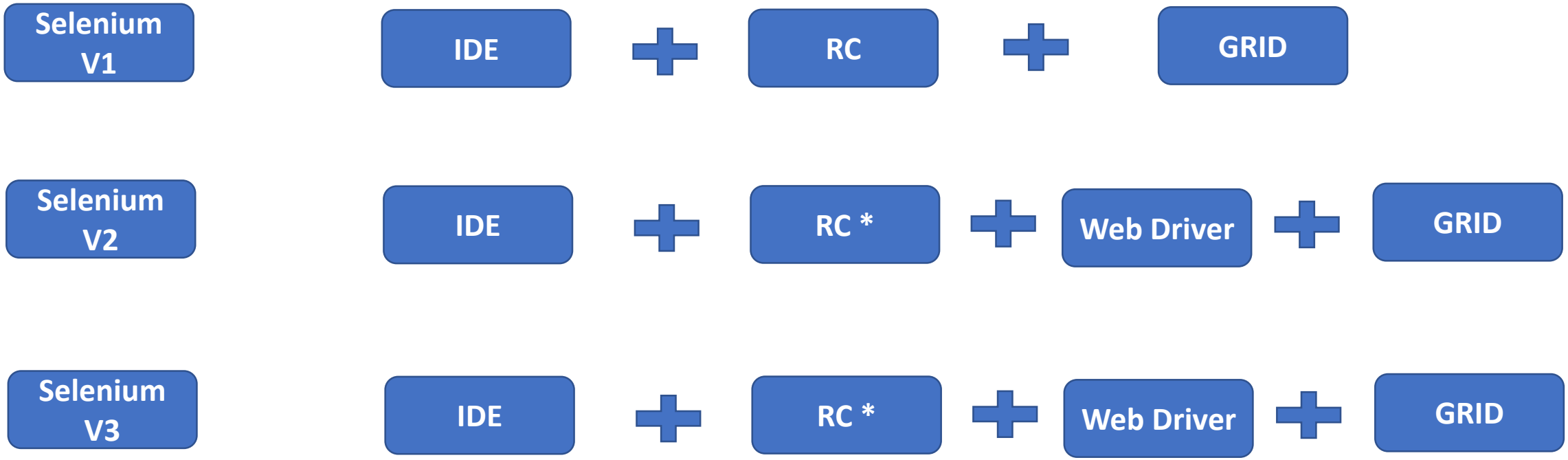
Product	Selenium	Katalon Studio	Unified Functional Testing	TestComplete	watir
Available since	2004	2015	1998	1999	2008
Application Under Test	Web apps	Web (UI & API), Mobile apps	Web (UI &API), Mobile, Desktop, Packaged apps	Web (UI &API), Mobile, Desktop apps	Web apps
Pricing	Free	Free	$$$$	$$	Free
Supported Platforms	Windows Linux OS X	Windows Linux OS X	Windows	Windows	Windows Linux OS X
Scripting languages	Java, C#, Perl, Python, JavaScript, Ruby, PHP	Java/Groovy	VBScript	JavaScript, Python, VBScript, JScript, Delphi, C++ and C#	Ruby
Programming skills	Advanced skills needed to integrate various tools	Not required. Recommended for advanced test scripts	Not required. Recommended for advanced test scripts	Not required. Recommended for advanced test scripts	Advanced skills needed to integrate various tools
Ease of Installation and Use	Require advanced skills to install and use	Easy to setup and use	Complex in installation. Need training to properly use the tool	Easy to setup. Need training to properly use the tool	Advanced skills needed to integrate various tools

Selenium

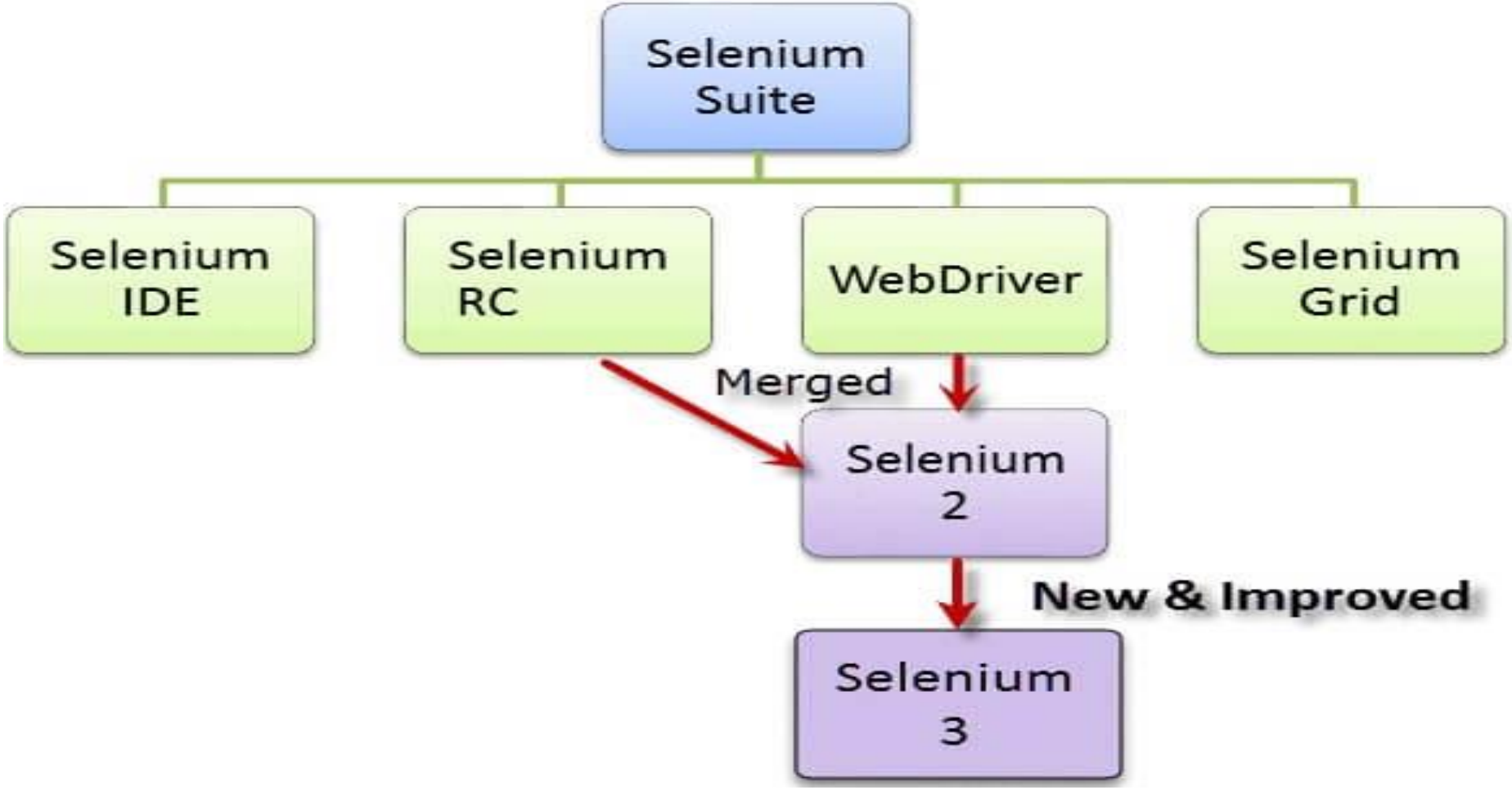
- Selenium is a free (open source) automated testing suite for web applications across different browsers and platforms. It is quite similar to HP Quick Test Pro (QTP now UFT) only that Selenium focuses on automating web-based applications. Testing done using Selenium tool is usually referred as Selenium Testing.
- Selenium is a free (open source) automated testing suite for web applications across different browsers and platforms. It is quite similar to HP Quick Test Pro (QTP now UFT) only that Selenium focuses on automating web-based applications. Testing done using Selenium tool is usually referred as Selenium Testing.
- 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.**
    1. Selenium Integrated Development Environment (IDE)
    2. Selenium Remote Control (RC)
    3. WebDriver
    4. Selenium Grid
-

Selenium Suite of tools

-



Selenium

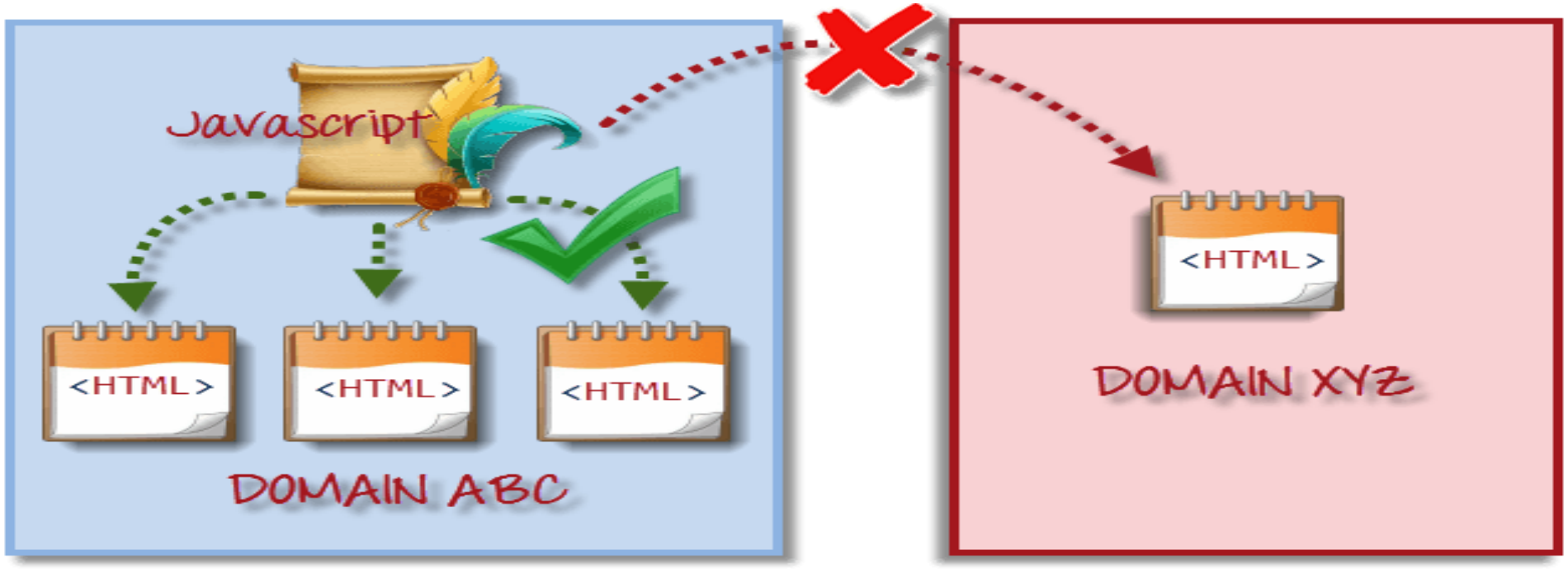

Selenium Suite
Selenium IDE
Selenium RC
WebDriver
Selenium Grid
Merged
Selenium 2
New & Improved
Selenium 3

Selenium - History

- Since Selenium is a collection of different tools, it had different developers as well. Below are the key persons who made notable contributions to the Selenium Project
- Primarily, Selenium was **created by Jason Huggins in 2004**. An engineer at ThoughtWorks, he was working on a web application that required frequent testing. Having realized that the repetitious Manual Testing of their application was becoming more and more inefficient, he created a JavaScript program that would automatically control the browser's actions. He named this program as the "**JavaScriptTestRunner**."
- Seeing potential in this idea to help automate other web applications, he made JavaScriptRunner open-source which was later re-named as Selenium Core.
-

Same origin policy

- **Same Origin policy prohibits JavaScript code from accessing elements from a domain that is different from where it was launched.**
-



under same origin policy, a Javascript program can only access pages on the same domain where it belongs. It cannot access pages from different domains

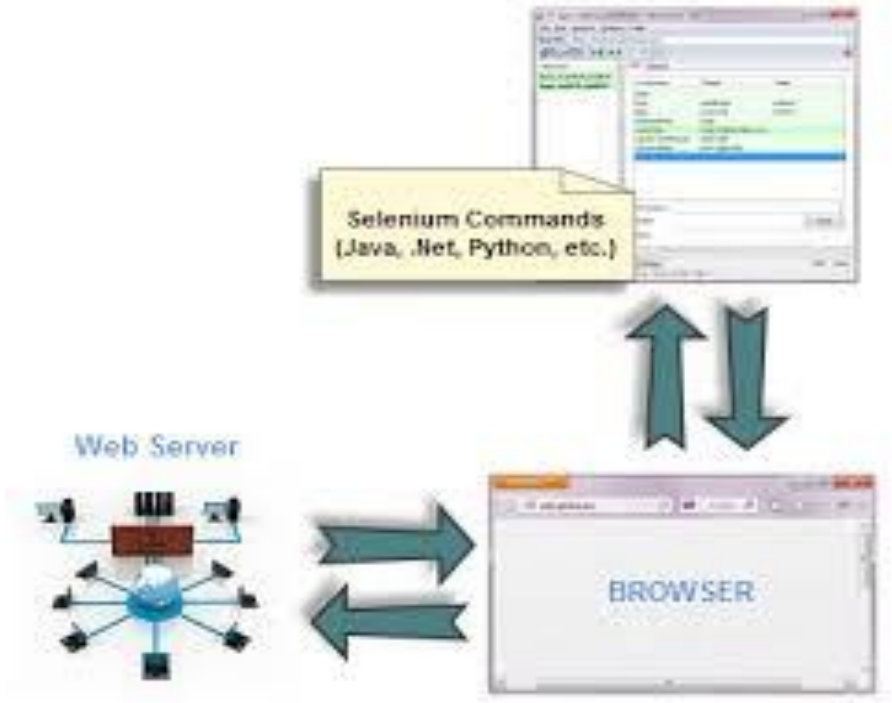
Birth of Selenium 2

- In **2008**, the whole Selenium Team decided to merge WebDriver and Selenium RC to form a more powerful tool called **Selenium 2**, with **WebDriver being the core**. Currently, Selenium RC is still being developed but only in maintenance mode. Most of the Selenium Project's efforts are now focused on Selenium 2.
- It came from a joke which Jason cracked one time to his team. Another automated testing framework was popular during Selenium's development, and it was by the company called **Mercury Interactive** (yes, the company who originally made QTP before it was acquired by HP). Since Selenium is a well-known antidote for Mercury poisoning, Jason suggested that name. His teammates took it, and so that is how we got to call this framework up to the present.

Birth of Remote Control (RC)

- Prior to Selenium RC, testers needed to install local copies of both **Selenium Core** (a JavaScript program) and the web server containing the web application being tested so they would belong to the same domain.
- **A**nother ThoughtWork's engineer, **Paul Hammant**, decided to create a server that will act as an HTTP proxy to "trick" the browser into believing that Selenium Core and the web application being tested come from the same domain. This system became known as the **Selenium Remote Control** or **Selenium 1**.
-

Remote Control (RC)



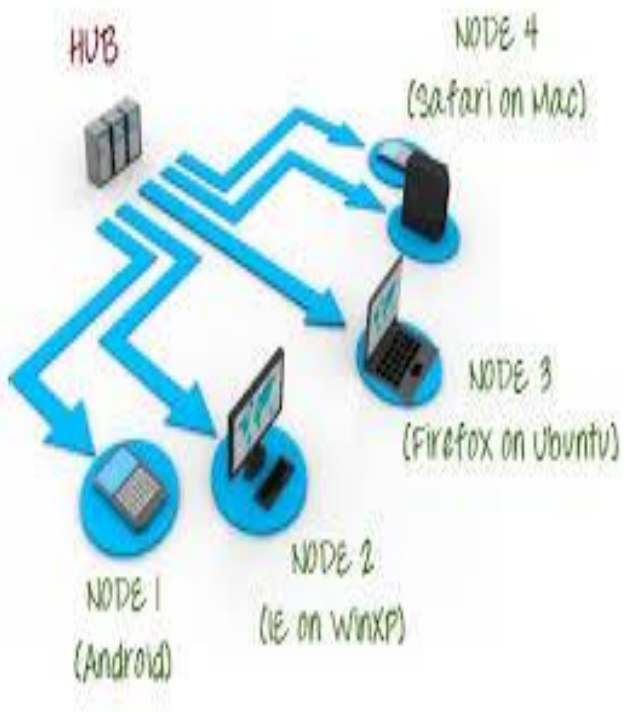
Working of RC

- First we need to launch **a separate application called Selenium Remote Control (RC) Server** before you can start testing
- The Selenium RC Server **acts as a "middleman" between your Selenium commands and your browser**
- When you begin testing, Selenium RC Server "injects" a **Javascript program called Selenium Core** into the browser.
- Once injected, Selenium Core will start receiving instructions relayed by the RC Server from your test program.
- When the instructions are received, **Selenium Core will execute them as Javascript commands.**
- The browser will obey the instructions of Selenium Core and will relay its response to the RC Server.
- The RC Server will receive the response of the browser and then display the results to you.
- RC Server will fetch the next instruction from your test script to repeat the whole cycle.
-

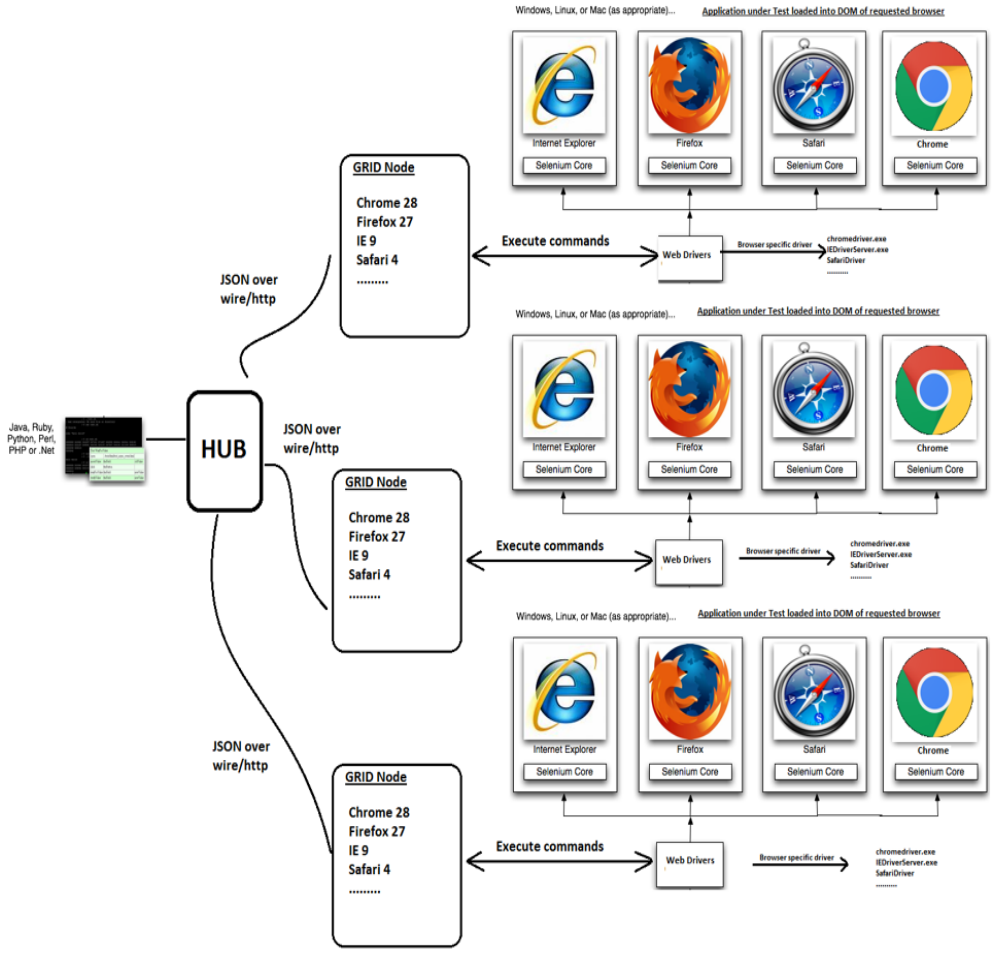
Birth of Selenium Grid

- Selenium Grid was developed by **Patrick Lightbody** to address the need of minimizing test execution times as much as possible. He initially called the system "**Hosted QA**." It was capable of capturing browser screenshots during significant stages, and also of **sending out Selenium commands to different machines simultaneously.**
-

Selenium Grid



Technology Stack – Infrastructure: Selenium Grid



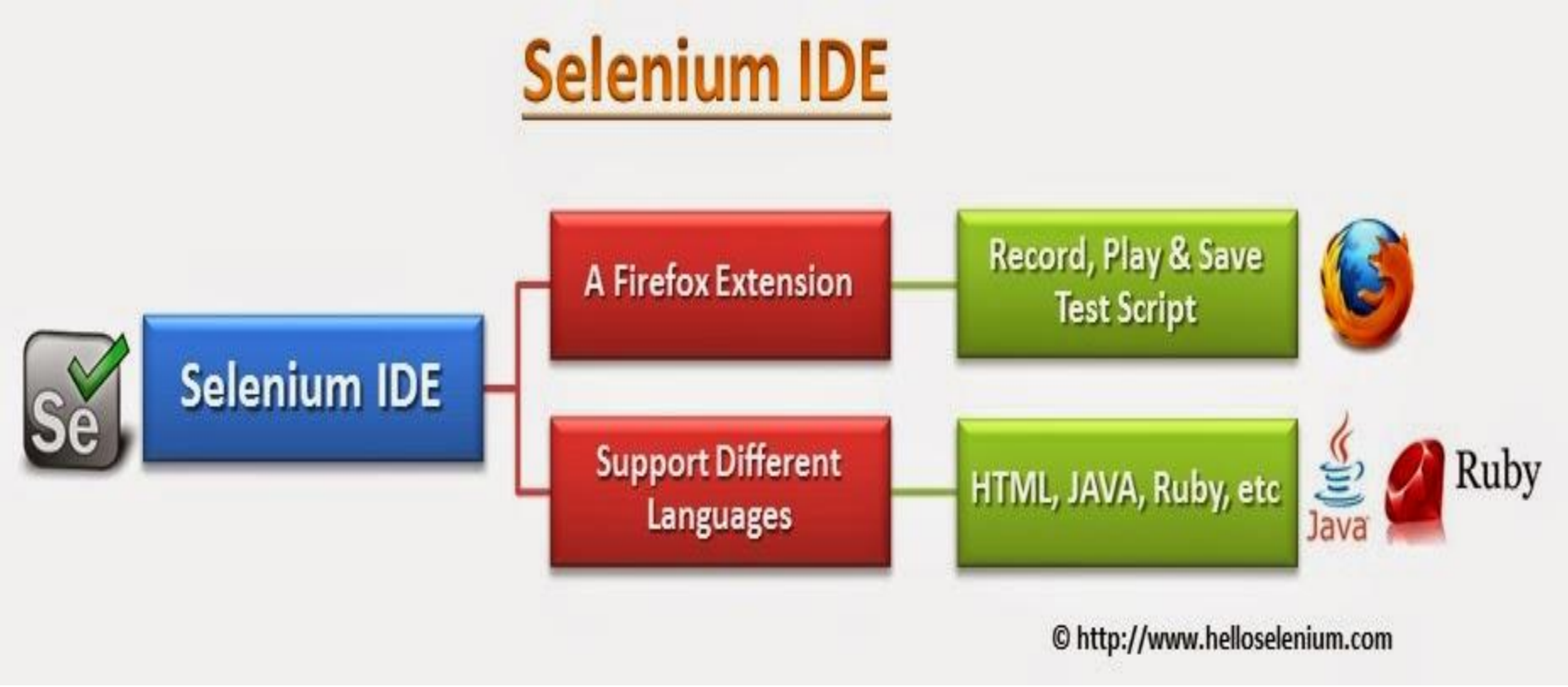
Selenium GRID

*Selenium GRID is a network of HUB & *Node. Each Nodes registers to the hub with a certain configuration and HUB is aware of the browsers available on the node. When a request comes to the HUB for a specific browser(with desidered capabilities object), the HUB, if found a match for the requested browser, re-directs the call to *thath* particular GRID Node and then a sessio is established bidirectionally and execution starts.*

Birth of Selenium IDE

- **Shinya Kasatani** of Japan created **Selenium IDE**, a Firefox extension that can automate the browser through a record-and-playback feature. He came up with this idea to further increase the speed in creating test cases. He donated Selenium IDE to the Selenium Project in **2006**.
-

Selenium IDE



Birth of Webdriver

- **Simon Stewart** created WebDriver circa **2006** when browsers and web applications were becoming more powerful and more restrictive with JavaScript programs like Selenium Core. **It was the first cross-platform testing framework that could control the browser from the OS level.**

Webdriver

- 
Workflow of Selenium Webdriver
Test Scripts
Webdriver
Browsers
Java
perl
Ruby
php
python
C#
Se


Mozilla FireFox
Opera Browser
Internet Explorer
WebDriver
GhostDriver
Safari
Google Chrome

Difference between RC and Webdriver

- WebDriver is a tool for testing web applications **across different browsers** using different programming languages.
- You are now able to make powerful tests because WebDriver **allows you to use a programming language** of your choice in designing your tests.
- WebDriver is **faster than Selenium RC** because of its simpler architecture.
- WebDriver **directly talks to the browser** while Selenium RC needs the help of the RC Server in order to do so.
- WebDriver's API is more **concise** than Selenium RC's.
- WebDriver **can support HtmlUnit** while Selenium RC cannot.
- The only drawbacks of WebDriver are:
- It **cannot readily support new browsers**, but Selenium RC can.
- It **does not have a built-in command** for automatic generation of test results.

Thank You

Raghu Prasad

9845547471

www.kaushalya.tech

FB : https://www.facebook.com/raghuprasadkonandur/

Skill Up-Skill Re-Skill

Excellence in action