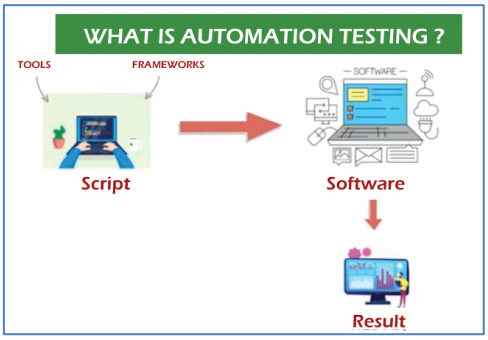
CHP 04 Test Automation

What is Automation Testing?

Automation Testing is a software testing technique that performs using special automated testing software tools to execute a test case suite. On the contrary, Manual Testing is performed by a human sitting in front of a computer carefully executing the test steps.

- It is used to automate the testing tasks that are difficult to perform manually.
- Automation tests can be run at any time of the day as they use scripted sequences to examine the software.
- Automation tests can also enter test data can compare the expected result with the actual result and generate detailed test reports.
- The goal of automation tests is to reduce the number of test cases to be executed manually but not to eliminate manual testing.
- It is possible to record the test suit and replay it when required.



Why Test Automation?

Test Automation is the best way to increase the effectiveness, test coverage, and execution speed in software testing. Automated software testing is important due to the following reasons:

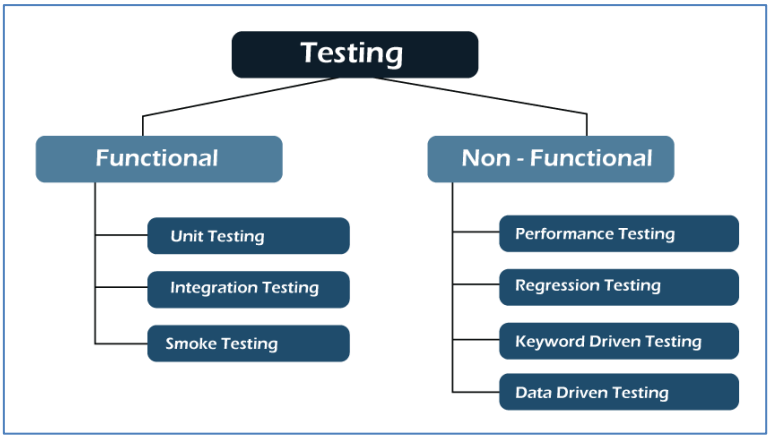
- Manual Testing of all workflows, all fields, all negative scenarios is time and money consuming
- It is difficult to test for multilingual sites manually
- Test Automation in software testing does not require Human intervention. You can run automated test unattended (overnight)
- Test Automation increases the speed of test execution
- Automation helps increase Test Coverage
- Manual Testing can become boring and hence error-prone.

Automation Testing Process

1. **Test Tool Selection**: There will be some criteria for the Selection of the tool. The majority of the criteria include: Do we have skilled resources to allocate for automation tasks, budget constraints, and Do the tool satisfies our needs?
2. **Define Scope of Automation**: This includes a few basic points such as the Framework should support Automation Scripts, Less Maintenance must be there, High Return on Investment, Not many complex Test Cases.
3. **Planning, Design, and Development:** For this, we need to Install particular frameworks or libraries, and start designing and developing the test cases such as NUnit, JUnit, QUnit, or required Software Automation Tools
4. **Test Execution**: Final Execution of test cases will take place in this phase and it depends on Language to Language for .NET, we'll be using NUnit, for Java, we'll be using JUnit, for JavaScript, we'll be using QUnit or Jasmine, etc.
5. **Maintenance**: Creation of Reports generated after Tests and that should be documented so as to refer to that in the future for the next iterations.

☯Automation testing types:

Testing is grouped under two types: functional and non-functional



Functional testing

The first test performed by tester on newly revised software is called functional testing, which verifies all the software functions' features per user requirement. This testing works on the real-world business application and obtaining the expected output from a given input. All application functions are tested and involve smoke, unit, and integration testing.

a) Unit testing:

The unit is the smallest component of the software that functions individually. Unit testing simplifies the testing of the whole software, where each software element is fully tested before the final version is out. Unit testing depicts how the code performs at each part and has a faster execution time.

b) Integration testing:

Integration testing is more complicated to set up compared with other tests. All the modules of the application communicate with each other to perform tasks. Therefore, testers group them for testing and exposing the flaws in maintaining the interaction between these modules.

c) Smoke testing

This testing checks and defines the product's stability (whether stable or not). If the product result is unstable, it is called an 'unstable build' and sent back to developers, where they run more test cases to find out the root cause of the problem.

Non-functional testing

Non-functional testing focuses on how well application functions are doing, not on what the product does. It is the opposite of functional testing, where application elements like reliability, usability, performance, etc., are tested. Some types of non-functional testing are reliability testing, load testing, compatibility testing, performance testing, security testing etc.

a) Performance testing

This non-functional testing tests the software's stability, responsiveness and speed under the workload. It finds out the potential issues faced by critical software and medical programs used by the user, like slow operation of software under stressful circumstances. It finds hurdles in the performance of software and removes them to increase the ability of software to deliver the best results to the end user.

b) Regression testing

When some changes are made to the code of software or application, it needs to be tested to determine whether the software is working as before the change; for this purpose, testers use automation regression testing to automate scripts, applications of workflows, plans and other activities. It tests the system or software workflow after its updation and functional error.

c) Keyword driven testing

Keyword-driven testing tests the application using the data files consisting of the keywords related to the application, representing a set of actions needed to carry out the step. Here these specific keywords are identified and connected with the specific action. Therefore, during testing, when these keywords are used, their related actions will automatically be done.

d) Data-driven testing

In data-driven testing, automation is inbuilt and very effective due to the few facilities provided, like the reusability of code, change in the script doesn't affect the test cases, and this testing can be carried out in the phase of the software development cycle. It provides consistency in results and reduces the investment of time and resources.

Advantages of Automation Testing:

Efficiency:

- Advantage: Automation testing is faster and more efficient than manual testing, especially when it comes to running repetitive, long, or complex test cases.
- Benefit: Faster test execution accelerates the feedback loop and allows for quicker identification and resolution of defects.

Reusability:

- Advantage: Test scripts can be reused across different test cycles, projects, or applications.
- Benefit: This reduces the effort required for creating new test cases and ensures consistent testing.

Consistency:

- Advantage: Automated tests execute the same steps and conditions consistently, reducing the chance of human error.
- Benefit: Consistency enhances the reliability of test results.

Parallel Execution:

- Advantage: Automation allows tests to run concurrently on different configurations, saving time and increasing test coverage.
- Benefit: Parallel execution is especially valuable for testing across various browsers, devices, or operating systems.

Regression Testing:

- Advantage: Automation is well-suited for regression testing, ensuring that existing functionality remains intact after code changes.
- Benefit: Regression tests can be executed quickly, providing immediate feedback to developers.

Improved Test Coverage:

- Advantage: Automation enables testing of a broader range of scenarios and configurations.
- Benefit: Enhanced test coverage increases the likelihood of detecting defects.

Early Bug Detection:

- Advantage: Automation can identify defects in the early stages of development.
- Benefit: Early detection allows for quicker resolution and cost savings in the long run.

Disadvantages of Automation Testing:

Initial Setup and Investment:

- Disadvantage: There is an initial investment required to set up automation testing, including tool selection, script development, and infrastructure.

Maintenance Overhead:

- Disadvantage: Automated tests need regular maintenance to adapt to changes in the application, which can be time-consuming and costly.

Not Suitable for All Testing:

- Disadvantage: Automation is not suitable for all types of testing, particularly for areas requiring human judgment, usability, or exploratory testing.

Limited Context Awareness:

- Disadvantage: Automated tests lack the contextual awareness and adaptability of human testers. They may not detect certain issues that require a human's understanding of the application's purpose.

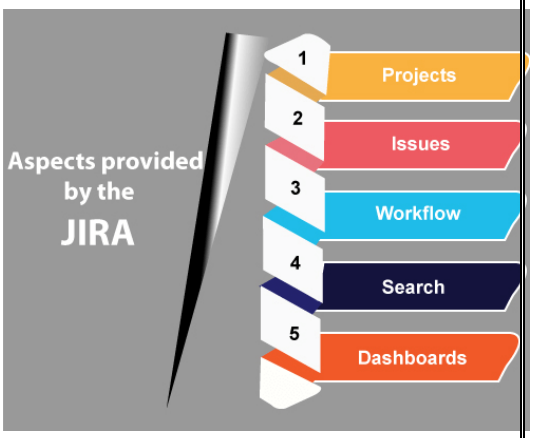
Tool Dependency:

- Disadvantage: Automation testing is tool-dependent, and selecting the right tool for your needs is crucial. It can be challenging to switch between automation tools.

Study of testing tools:

✡What is JIRA?

JIRA is a tool developed by Australian Company Atlassian. This software is used for bug tracking, issue tracking, and project management. The JIRA full form is actually inherited from the Japanese word “Gojira” which means “Godzilla”. The basic use of this tool is to track issue and bugs related to your software and Mobile apps.



- **JIRA** is based on the Agile methodology and the current version of the Jira is 6.
- **Projects**: It is used to manage the defects very effectively.
- **Issue**: It is used to track and manage the defects/issues.

- **Workflow**: Processes the Issue/Defect life cycle. Suppose we have a business requirement, we create the technical design and from the technical design, we create the test cases. After creating the test cases, coding is done, and then testing is performed on the project. This design workflow is possible by using Jira.

- **Search**: Find with ease. Suppose we have done with a project at the beginning of the December and its version is 1.0. Now, we move to version 1.1 and completed at the end of December. What we are doing is that we are adding new versions. Through Jira, we can get to know that what happened in the earlier versions, how many defects occurred in the earlier projects and the learning we achieve from the earlier projects.

- **Dashboards**: Dashboard is a display which you see when you log in to the Jira. You can create multiple dashboards for multiple projects. You can create the personal dashboard and can add the gadgets in a dashboard so that you can keep track of the assignments and issues that you are working on.

How to Use JIRA?

Here is a step by step process on how to use Jira software:

- Step 1) Open Jira software and navigate to the Jira Home icon
- Step 2) Select Create project option
- Step 3) Choose a template from the library
- Step 4) Set up the columns as per your need from Board settings
- Step 5) Create an issue
- Step 6) Invite your Team members and start working.

❤️Why JIRA

JIRA tool is used because of the following reasons:

- **Issue Tracking**: JIRA allows teams to create, track, and manage issues and tasks. This is particularly useful for software development, where issues can range from bugs to new feature requests.
- **Project Management**: JIRA provides tools for project planning, task assignment, and progress monitoring. Teams can create and manage sprints, epics, and user stories, making it a valuable tool for agile project management.
- **Customization**: JIRA is highly customizable. Teams can define their own issue types, workflows, fields, and notifications to match their specific project needs.
- **Collaboration**: It fosters collaboration by enabling teams to comment on issues, attach files, and link issues together. This is essential for communication and knowledge sharing among team members.
- **Integration**: JIRA integrates seamlessly with a wide range of other development and collaboration tools. It can be integrated with version control systems, continuous integration tools, and documentation platforms.
- **Reporting** and **Dashboards**: JIRA offers reporting features that allow teams to create custom reports and dashboards to track progress, identify bottlenecks, and make data-driven decisions.
- **Scalability**: JIRA can scale to accommodate small teams or large enterprises. It's suitable for various industries and types of projects.
- **Agile Support**: It's widely used for agile development methodologies, such as Scrum and Kanban. JIRA provides agile boards, backlogs, and burndown charts.
- **Security**: JIRA offers robust security features, allowing administrators to manage user permissions and access control.

Advantages of JIRA:

- **Flexible and Customizable**: JIRA is highly customizable and can be adapted to suit the specific needs of your team or organization. You can create custom workflows, issue types, and fields to match your project's requirements.
- **Scalable**: It can scale to accommodate the needs of both small and large teams and can handle complex projects with a large number of issues.
- I**ntegration**: JIRA seamlessly integrates with other Atlassian products like Confluence, Bitbucket, and Trello, as well as numerous third-party tools and apps through the Atlassian Marketplace.
- **Effective Issue Tracking**: It excels at issue and bug tracking, helping teams identify, prioritize, and resolve issues efficiently.
- **Real-time Collaboration**: JIRA provides real-time collaboration features that allow team members to comment, share information, and collaborate on tasks and issues.

Disadvantages:

- **Complexity**: JIRA can be overwhelming for new users and smaller teams. Its extensive customization options can sometimes lead to over-complexity.
- **Cost**: JIRA can be expensive, especially for larger teams or organizations. The cost may include licensing fees, add-ons, and the need for dedicated administrators.
- **Learning Curve**: Due to its richness of features and complexity, it can have a steep learning curve for new users. Training and onboarding may be necessary.
- **Resource-Intensive**: Running JIRA on self-hosted servers can be resource-intensive, and performance issues may arise if not properly managed.**:**

☢What is Bugzilla?

- Bugzilla is a bug tracking tool that helps to track the issues related to their product.
- Bugzilla tool is written in Perl language, and it uses MySQL database.
- It is a bug tracking tool. However, it can also be used as a test management tool because it can be linked with other Test case management tools such as Quality Center, Testlink, etc.
- It is an open source tool, i.e., this tool is available to the users at a free of cost.

Key features of Bugzilla:

Advanced Search capabilities

- It has Google-like bug search which is simple to use, and it also searches the full text of a bug.
- It provides you a very advanced search system where you can create any type of search that you want such as time-based searches (For example, you want to see the list of bugs whose priority has been changed since last two days).

Email Notifications controlled by user preferences

- You will get an email if any changes are made in the Bugzilla, and the notifications that you get on which bug is totally controlled by the user preferences.

ug lists in multiple formats

Reports and Charts When you search for the bugs, then you can get the bug lists in multiple formats such as Atom, iCalendor format. The iCalendor format is used when you are using the time tracking feature in Bugzilla. There are even more formats available in Bugzilla such as printable format that contains the details of all the bugs, CSV format used for importing bug list into spreadsheets.

Scheduled reports by email

Bugzilla has a system that will send you, users or a group that you specify the results of a search on a schedule that you have mentioned.

Automatic Duplicate Bug Detection

When you are filing a new bug, and when you type the summary for the bug, then the system looks for similar bugs. If the system finds the similar bugs, then it allows the user to add themselves in the CC list of one of those bugs instead of creating a new one.

File/Modify Bugs by email

You can send an email to Bugzilla to create a new bug or modify the existing bug. You can even attach the files to Bug.

Time Tracking

Bugzilla also provides the feature of time tracking. You can determine how many hours a bug will take to get fixed and you can also even track the hours that you need to spend on the bug. You can also set the deadline by which the bug needs to be fixed.

☸What is Selenium

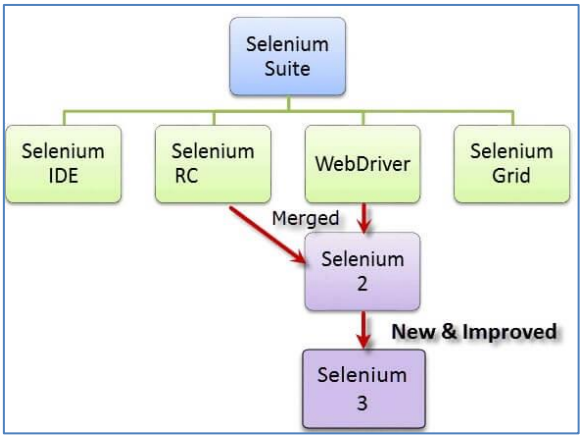
- Selenium is one of the most widely used open source Web UI (User Interface) automation testing suite.
- It was originally developed by Jason Huggins in 2004 as an internal tool at Thought Works.
- Selenium supports automation across different browsers, platforms and programming languages.
- Selenium can be easily deployed on platforms such as Windows, Linux, Solaris and Macintosh. Moreover, it supports OS (Operating System) for mobile applications like iOS, windows mobile and android.

- Selenium supports a variety of programming languages through the use of drivers specific to each language.Languages supported by Selenium include C#, Java, Perl, PHP, Python and Ruby.Currently, Selenium Web driver is most popular with Java and C#.
- Selenium test scripts can be coded in any of the supported programming languages and can be run directly in most modern web browsers. Browsers supported by Selenium include Internet Explorer, Mozilla Firefox, Google Chrome and Safari.

✡Selenium Tool Suite:

Selenium Software is not just a single tool but a suite of software, each piece catering to different Selenium QA testing needs of an organization. Here is the list of tools

- **Selenium Integrated Development Environment (IDE)**
- **Selenium Remote Control (RC)**
- **WebDriver**
- **Selenium Grid**



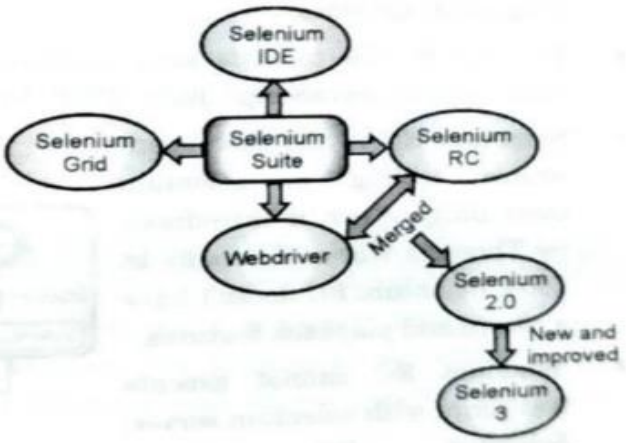
4.6.3 Selenium IDE

UQ. What is Selenium's IDE explain in detail. **MU - Oct. 18**

- Selenium IDE (Integrated Development Environment) is an open source web automation testing tool under the Selenium Suite. Unlike Selenium WebDriver and RC, it does not require any programming logic to write its test scripts rather you can simply record your interactions with the browser to create test cases. Subsequently, you can use the playback option to re-run the test cases.
- Selenium Integrated Development Environment (IDE) is the simplest framework in the Selenium suite. It is a browser plugin to record and playback the operations performed on the browser. Selenium IDE plugins are available for Chrome and Firefox browsers. It doesn't support the programming features. Selenium is the language which is used to write test scripts in Selenium IDE.

As a Firefox plugin, Selenium Integrated Development Environment (IDE) can be used to create a test script prototype quickly and easily. It can record human testers' actions as a script while the tester runs the test case manually. Selenium IDE is a rapid prototyping tool for building test scripts within very less amount of time. It allows you to record, edit and debug the test case by providing the very simple to use components. This tool will be most helpful for beginners to learn the commands used by selenium while recording the test case. Although it was available as Firefox add-on

- The recorded test script can be executed at a later point in time for the regression test automatically. This tool can access the browser's DOM elements with the use of JavaScript.
- It also provides a flexible interface for testers to create or update test cases. Thought Works Company introduces selenium IDE in 2006 andimplemented in the Firefox browser, which provides record and playback functionality to the test scripts.
- Selenium-IDE is the simplest tool of Selenium community. Selenium-IDE allows software testers to export recorded scripts in many languages like HTML, Java, Ruby, PHP, Python, C#, and Test-NG.
- Selenium–IDE supports six locators, i.e., - Id, Name, X Path, CSS Selector, Link Text, DOM.



(104)Fig. 4.6.2 : Selenium Suite

(104)Fig. 4.6.2 : Selenium Suite

Pros

1. It is simple, easy to install and use.
2. Built-in test results reporting and help modules.
3. Test Cases can be exported to usable formats in the Selenium RC and WebDriver.

Cons

1. It is only available for Firefox.
2. The execution of test cases is slow as compared to RC and WebDriver.
3. Data-driven testing is not supported.
4. It is not able to test dynamic web applications.

4.6.4 Selenium RC

UQ. What is Selenium's RC explain in detail. **MU - May 18**

- Selenium RC is the main feature in the Selenium. A tester can use it to simulate user actions such as input data, submit a form, and click a button in web browsers.
- Selenium RC was the first tool used on selenium project. It was the core application written in Java as a programming language.
- This tool will accept commands for the browser via HTTP request. It consists of two components which are selenium RC server and RC client. Where RC server will communicate with HTTP/GET/POST request while the RC client will include programming codes.
- You will be able to write an automated test in most of the programming languages such as Java, JavaScript, Ruby, PHP, Python, and Perl and C #.

such as Java, JavaScript, Ruby, PHP, Python, and Perl and C #.

- Selenium RC is the first open-source tooling in selenium community which is introduced by Thought Works Company in 2004. Selenium RC doesn't have a record and playback features.
- Selenium RC cannot execute test script with selenium server. Selenium –RC supports multiple languages (java, C++, python), multiple operating systems (Windows, Linux), and multiple browsers (internet explorer, Google Chrome).

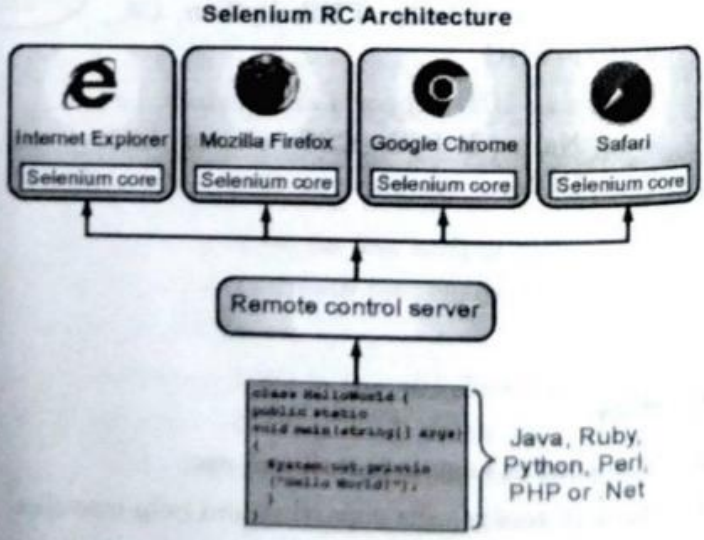


Fig. 4.6.3 : Core Components of Selenium RC

Usage of Selenium RC

1. Tester writes a test case script with the supported programming language API.
2. The test script sends a command to the RC server.
3. RC server receives these commands and triggers selenium core from executing the commands and interacting with the browser page web elements.

Pros

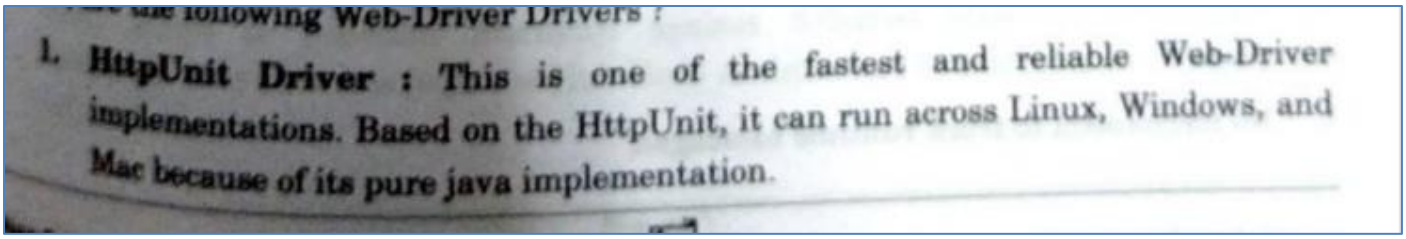
1. It supports cross-browser testing.
2. It supports data-driven testing.
3. Execution speed is more as compared to IDE.
4. It supports conditional operations and iterations.

Cons

1. Slower execution speed as compared to Web-Driver.
2. Browser interaction is less realistic.
3. Programming knowledge required.

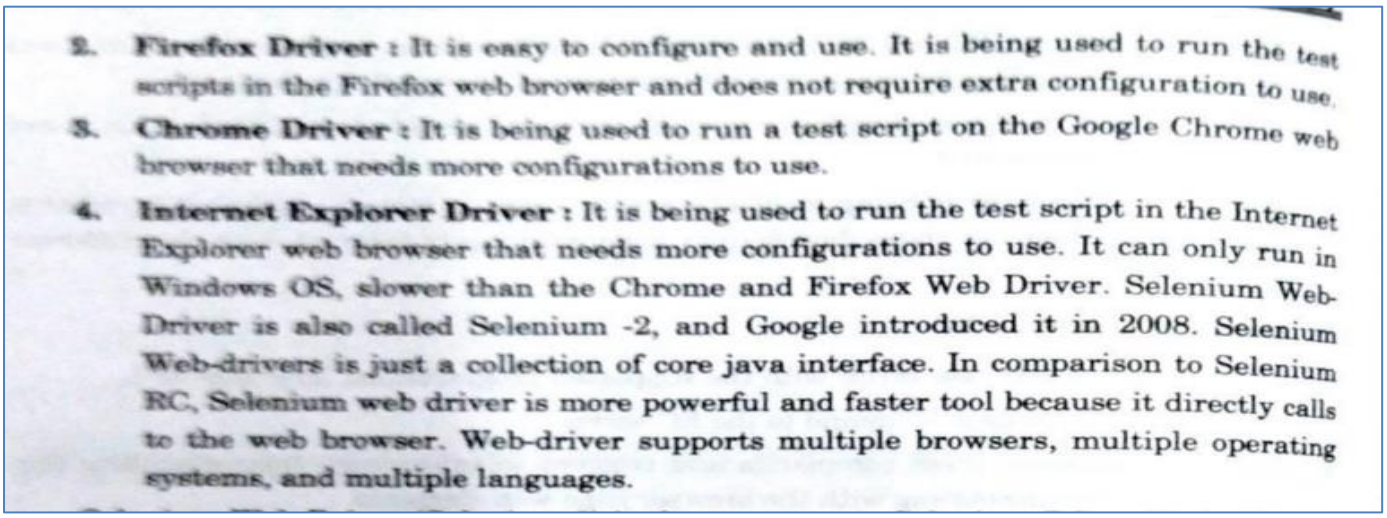
☯What is WebDriver?

The WebDriver proves to be better than Selenium IDE and Selenium RC in many aspects. It implements a more modern and stable approach in automating the browser's actions. WebDriver, unlike Selenium RC, does not rely on JavaScript for Selenium Automation Testing. It controls the browser by directly communicating with it.

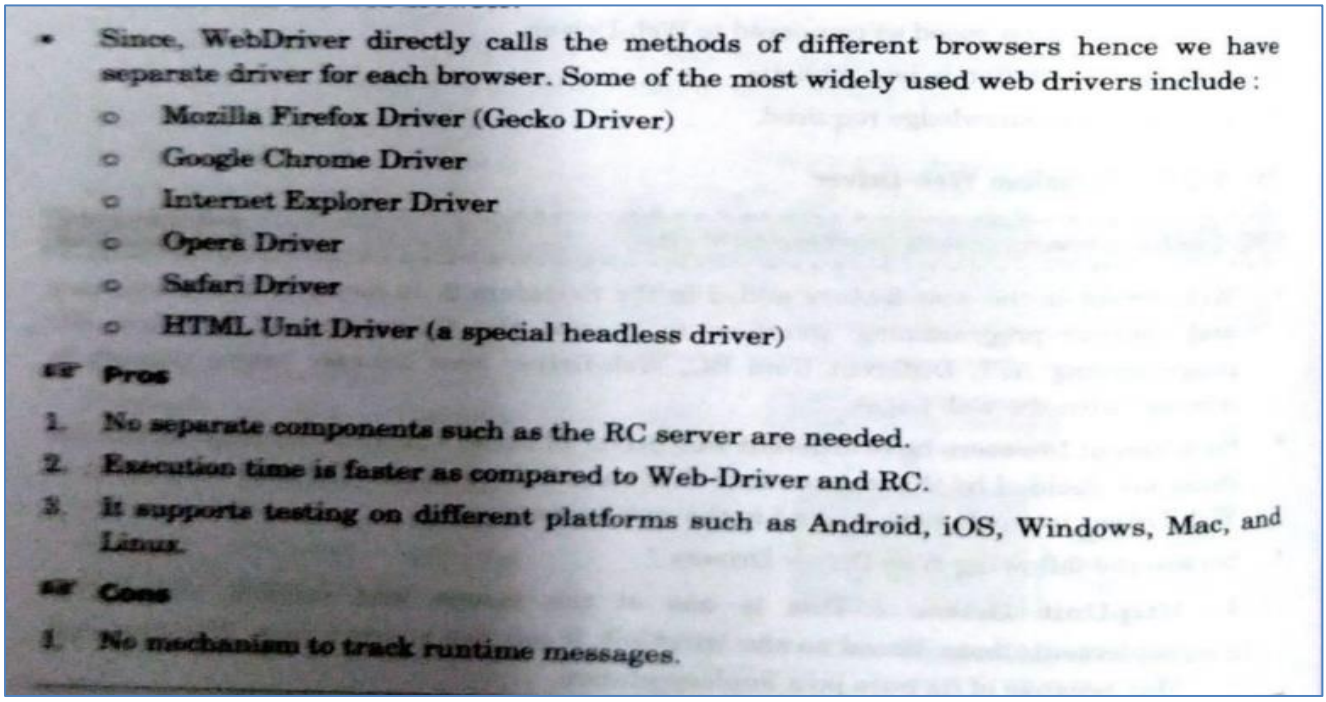


the following Web-Driver Drivers :

1. **HttpUnit Driver :** This is one of the fastest and reliable Web-Driver implementations. Based on the HttpUnit, it can run across Linux, Windows, and Mac because of its pure java implementation.



2. **Firefox Driver :** It is easy to configure and use. It is being used to run the test scripts in the Firefox web browser and does not require extra configuration to use.
3. **Chrome Driver :** It is being used to run a test script on the Google Chrome web browser that needs more configurations to use.
4. **Internet Explorer Driver :** It is being used to run the test script in the Internet Explorer web browser that needs more configurations to use. It can only run in Windows OS, slower than the Chrome and Firefox Web Driver. Selenium Web-Driver is also called Selenium -2, and Google introduced it in 2008. Selenium Web-drivers is just a collection of core java interface. In comparison to Selenium RC, Selenium web driver is more powerful and faster tool because it directly calls to the web browser. Web-driver supports multiple browsers, multiple operating systems, and multiple languages.



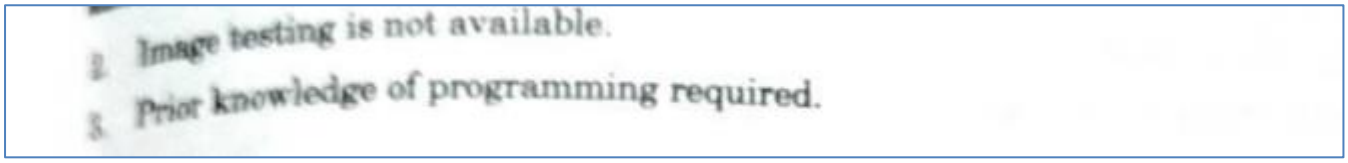
- Since, WebDriver directly calls the methods of different browsers hence we have separate driver for each browser. Some of the most widely used web drivers include :
  - Mozilla Firefox Driver (Gecko Driver)
  - Google Chrome Driver
  - Internet Explorer Driver
  - Opera Driver
  - Safari Driver
  - HTML Unit Driver (a special headless driver)

Pros

1. No separate components such as the RC server are needed.
2. Execution time is faster as compared to Web-Driver and RC.
3. It supports testing on different platforms such as Android, iOS, Windows, Mac, and Linux.

Cons

1. No mechanism to track runtime messages.



2. Image testing is not available.
3. Prior knowledge of programming required.

GQ. What is Test Director ?

- It is a Global Test Management tool, the industry's first global test management solution.

- It helps organizations deploy high-quality applications more quickly and effectively.
- It has four modules:
  1. Requirements
  2. Test Plan
  3. Test Lab
  4. Defects
- These modules are seamlessly integrated, allowing for a smooth information flow between various testing stages. The completely Web-enabled TestDirector supports high levels of communication and collaboration among distributed testing teams, driving a more effective, efficient global application-testing process.
- **Web-based Site Administrator :** The Site Administrator includes tabs for managing projects, adding users and defining user properties, monitoring connected users, monitoring licenses and monitoring TestDirector server information.

- **Domain Management :** TestDirector projects are now grouped by domain. A domain contains a group of related TestDirector projects, and assists you in organizing and managing a large number of projects. Enhanced Reports and Graphs Additional standard report types and graphs have been added, and the user interface is richer in functionality. The new format enables you to customize more features. Version Control Version control enables you to keep track of the changes you make to the testing information in your TestDirector project. You can use your version control database for tracking manual, WinRunner and QuickTest Professional tests in the test plan tree and test grid.
- **Collaboration Module :** The Collaboration module, available to existing customers as an optional upgrade, allows you to initiate an online chat session with another TestDirector user. While in a chat session, users can share applications and make changes.
- **TestDirector Advanced Reports Add-in :** With the new Advanced Reports Add-in, TestDirector users are able to maximize the value of their testing project information by generating customizable status and progress reports. The Advanced Reports Add-in offers the flexibility to create custom report configurations and layouts, unlimited ways to aggregate and compare data and ability to generate cross-project analysis reports.
- **Automatic Traceability Notification :** The new traceability automatically traces changes to the testing process entities such as requirements or tests, and notifies the user via flag or e-mail. For example, when the requirement changes, the associated test is flagged and the user is notified that the test may need to be reviewed to reflect

- **Workflow for all TestDirector Modules :** The addition of the script editor to all modules enables organizations to customize TestDirector to follow and enforce any methodology and best practices. Improved Customization With a greater number of available user fields, ability to add memo fields and create input masks users can customize their TestDirector projects to capture any data required by their testing process. New rich edit option add color and formatting options to all memo fields.
- TestDirector Features and Benefits Supports the entire testing process. TestDirector incorporates all the following aspects of the testing process into a single browser-based application :
  - Requirements management
  - Planning
  - Scheduling
  - Running tests
  - Issue management
  - Project status analysis
- Leverages innovative Web technology Testers, developers and business analysts can participate in and contribute to the testing process by working seamlessly across geographic and organizational boundaries.
- **Uses industry-standard repositories :** TestDirector integrates easily with

- **Generates customizable reports :** TestDirector features a variety of customizable graphs and reports that provide a snapshot of the process at any time during testing. You can save your favorite views to have instant access to relevant project information.
- **Supports decision-making through analysis :** TestDirector helps you make informed decisions about application readiness through dozens of reports and analysis features. Provides Anytime, Anywhere Access to Testing Assets Using TestDirector's Web interface, testers, developers and business analysts can participate in and contribute to the testing process by collaborating across geographic and organizational boundaries.
- **Provides Traceability :** Throughout the Testing Process: TestDirector links requirements to test cases, and test cases to issues, to ensure traceability throughout the testing cycle. When requirement changes or the defect is fixed, the tester is notified of the change.
- **Integrates with Third-Party Applications :** Whether an individual uses an industry standard configuration management solution, Microsoft Office or a homegrown defect management tool, any application can be integrated into TestDirector. Through the open API, TestDirector preserves the users' investment in their existing solutions and enables them to create an end-to-end lifecycle-management solution.