**1. What is Selenium and what are the different components and versions of Selenium?**

**Answer:**Selenium is a free (open source) automated testing suite for web applications across different browsers and platforms. 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.

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

**Selenium Core**

The story starts in 2004 at ThoughtWorks in Chicago, with Jason Huggins building the Core mode as "JavaScriptTestRunner“. Its JavaScript program that would automatically control the browser's actions.

“JavaScriptTestRunner” was later named as “Selenium Core” and released into the market as an Open Source tool. This Open Source tool started gaining demand in the market and people started using it for automating the repeated tasks in their Web Applications.

**Selenium Remote Control**

Unfortunately; testers using Selenium Core had to install the whole application under test and the web server on their own local computers because of the restrictions imposed by the same origin policy. To resolve this another ThoughtWork's engineer, Paul Hammant created system (in 2007) known as the Selenium Remote Control or Selenium 1.

Selenium 1 = Selenium IDE + Selenium RC + Selenium Grid

**Selenium Grid**

Patrick Lightbody to address the need of minimizing test execution times as much as possible, So he created Selenium Grid. Basically grid is for parallel execution and execute your test scripts on multiple environments.

Using “Selenium Grid”, testers were able to distribute the tests across multiple machines and get them executed them on different machines over their network to reduce or minimize the time taken for overall execution of tests.

**Selenium IDE**

“Shinya Kasatani”, who developed a Firefox extension named as “Selenium IDE”.

“Selenium IDE” using its record and playback feature, records the automation tests like recording a video and executes the recorded tests like playing the recorded videos.

**Selenium WebDriver**

Earlier Selenium 1 used to be the major project of Selenium.

Selenium 1 = Selenium IDE + Selenium RC + Selenium Grid

Later Selenium Team has decided to merge both Selenium WebDriver and Selenium RC to form a more powerful Selenium tool.

They both got merged to form “Selenium 2”

“Selenium WebDriver” was the core of “Selenium 2” and “Selenium RC” used to run in maintenance mode.

Hence Selenium 2 = Selenium IDE + Selenium WebDriver 2.x + Selenium Grid.

“Selenium 2” released on July 8, 2011. Selenium team has decided to completely remove the dependency for Selenium RC. After 5 years, “Selenium 3" was released on October 13, 2016 with a major change, which is the original Selenium Core implementation and replacing it with one backed by WebDriver and lot more improvements.

Hence Selenium 3 = Selenium IDE + Selenium WebDriver 3.x + Selenium Grid.

After 3 years from it’s a major release, now Selenium has put out its first alpha version of Selenium 4 on Apr 24, 2019. Still, there is no official announcement about the release date of Selenium 4, but we are expecting it around October 2019. Till that there can be several alpha or beta versions released time to time with stabilization.

**2. What is the latest Selenium tool?**

**Answer:** Selenium WebDriver is the successor to Selenium RC which sends commands directly to the browser and retrieves results. Selenium Grid is a tool used to run parallel tests across different machines and different browsers simultaneously which results in minimized execution time.

**3. What is the difference between Manual and Automation Testing?**

**Answer:**Differences are given below:

|  |  |
| --- | --- |
| **Automation Testing** | **Manual Testing** |
| Automated testing is more reliable. It performs same operation each time. It eliminates the risk of human errors. | Manual testing is less reliable. Due to human error, manual testing is not accurate all the time. |
| Automation Testing uses automation tools to execute test cases. | In manual testing, test cases are executed by a human tester and software. |
| Initial investment of automation testing is higher. Investment is required for testing tools. In the long run it is less expensive than manual. ROI is higher in the long run compared to Manual testing. | Initial investment of manual testing is less than automation. Investment is required for human resources. ROI is lower in the long run compared to Automation testing. |
| Automation testing is a practical option when we do regressions testing. | Manual testing is a practical option where the test cases are not run repeatedly and only needs to run once or twice. |
| Automation does not allow random testing | Exploratory testing is possible in Manual Testing |
| Automated testing is a reliable method, as it is performed by tools and scripts. There is no testing Fatigue. Manual testing is not as accurate because of the possibility of the human errors. | Manual testing is not as accurate because of the possibility of the human errors. |
| Automated testing does not involve human consideration. So it can never give assurance of user-friendliness and positive customer experience. | The manual testing method allows human observation, which may be useful to offer user-friendly system. |
| Performance Tests like Load Testing, Stress Testing, Spike Testing, etc. have to be tested by an automation tool compulsorily. | Performance Testing is not feasible manually |
| This testing can be executed on different operating platforms in parallel and reduce test execution time. | Manual tests can be executed in parallel but would need to increase your human resource which is expensive |
| Automation testing uses frameworks like Data Drive, Keyword, Hybrid to accelerate the automation process. | Manual Testing does not use frameworks but may use guidelines, checklists, stringent processes to draft certain test cases. |
|  |  |

**4. What are the benefits of Automation Testing?**

**Answer:** Following are benefits of automated testing:

* 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.

**5. What are the popular test automation tools for functional testing?**

**Answer:** Please follow below link from “softwaretestinghelp.com”  . It is very good article on different testing tools.

<https://www.softwaretestinghelp.com/tools/top-30-functional-testing-tools/>

**6. What is the main purpose of Automation Testing?**

**Answer:**Automated software testing can increase the depth and scope of tests to help improve software quality. Lengthy tests that are often avoided during manual testing can be run unattended. They can even be run on multiple computers with different configurations.

Test Automation demands considerable investments of money and resources.

Successive development cycles will require execution of same test suite repeatedly. 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. The goal of Automation is to reduce the number of test cases to be run manually and not to eliminate Manual Testing altogether.

**7. What is the goal of Automation Testing?**

**Answer:**The goal of Automation is to reduce the number of test cases to be run manually and not to eliminate Manual Testing altogether.

**8. Why Selenium should be selected as a Test tool?**

**Answer:**Benefits of selecting Selenium:

It is completely open source - So you can easily download it for FREE. Selenium supports different programming languages such as Java, Python, C#, Ruby, Groovy, Javascript etc - So you can easily use Selenium. Selenium works in multiple operating systems - Hence no worries about using it.

**9. What is Same Origin Policy and how it can be handled? How to overcome same origin policy through web driver?**

**Answer:** Same Origin policy prohibits JavaScript code from accessing elements from a domain that is different from where it was launched. Example, the HTML code in www.google.com uses a JavaScript program "testScript.js". The same origin policy will only allow testScript.js to access pages within google.com such as google.com/mail, google.com/login, or google.com/signup. However, it cannot access pages from different sites such as yahoo.com/search or fbk.com because they belong to different domains.

**10. What are the testing types that can be supported by Selenium?**

**Answer:** Using Selenium type of testing can be done are:

* Functional Testing.
* Regression Testing.
* Sanity Testing.
* Smoke Testing.
* Responsive Testing.
* Cross Browser Testing.
* UI testing (black box)
* Integration Testing.

**11. What are the limitations of Selenium?**

**Answer:**Challenges and limitations of Selenium WebDriver.

* We cannot test windows application.
* We cannot test mobile apps.
* Limited reporting.
* Handling dynamic Elements.
* Handling page load.
* Handling pop up windows.
* Handling captcha.

**12. Which Test cases needs to be automated?**

**Answer:**test cases can (and should be automated) if:

* Tests are used repeatedly.
* Tests involve a lot of data entry.
* Tests clearly pass or fail.
* Tests deliver an exact result.
* Tests use consistent UI and regular controls.
* Tests are only to do what they're told — not check anything else.

**13. What is Selenese?**

**Answer:** Selenese is the set of selenium commands which are used to test your web application. Tester can test the broken links, existence of some object on the UI, Ajax functionality, Alerts, window, list options and lot more using selenese. Selenium command tells selenium automation engine to perform certain tasks.

**14. Which automation tools could be used for post-release validation with continuous integration?**

**Answer:**

1) **Experitest**- Experitest is the continuous testing platform for mobile and web apps. It integrates seamlessly with your development, testing, and continuous integration eco-system, and is fully compatible with Appium, Selenium, Jenkins and other tools.

2) **Selenium**

Selenium is open-source software testing tool. It supports all the leading browsers like Firefox, Chrome, IE, and Safari. Selenium WebDriver is used to automate web application testing.

3) **QuerySurge**

QuerySurge is the smart data testing solution that is the first-of-its-kind full DevOps solution for continuous data testing.

Key features include detailed data intelligence & data analytics, seamless integration into the DevOps pipeline for continuous testing.

4) **Jenkins**

Jenkins is a Continuous Integration tool which is written using Java language. This tool can be configured via GUI interface or console commands.

5) **Travis**

Travis is continuous testing tool hosted on the GitHub. It offers hosted and on-premises variants. It provides a variety of different languages and a good documentation.

**15. Does the latest version of Selenium WebDriver support Mobile Testing?**

**Answer:** No. Selenium cannot automate Mobile Applications.you should use Appium for mobile automation.

**16. Explain your project, roles & Experience summary?**

**Answer:**Following are the responsibilities according to the experience level:

1 to 2 Years

==========

* Understanding Test Requirements and analyzing the Application under Test(AUT).
* Generating Test Cases (Test Scripts) using Selenium Element locators and WebDriver API Commands.
* Enhancing Test cases using Java Programming.
* Debugging Test Cases and Fixing Errors.
* Executing/Running Test Cases
* Defect Reporting & Tracking
* Test Reporting

2 to 4  years of Experience in Test Automation using Selenium

==========================================================

* Understanding and Analyzing the Application Under Test in terms of Object Identification.
* Creating Test scenarios and Collecting Test Data.
* Creating Test Automation Resources (Function Libraries etc…).
* Implementing JUnit or TestNG Test Automation framework and developing automation infrastructure.
* Creating and enhancing Test Cases (Test Scripts) using Element locators, WebDriver methods, Java programming concepts and TestNG Annotations.
* Creating reusable components.
* Handling duplicate objects and dynamic objects using index property and Regular expressions.
* Collecting Test Data for Data Driven Testing.
* Creating Test Cases (Test Scripts) using Selenium Webdriver, Java and TestNG Annotations.
* Parameterization, Synchronization and define Test results.
* Error Handling, Adding comments.
* Creating Data driven Tests and Running through framework.
* Debugging and Running Tests
* Analyzing Test Results
* Defect Reporting and Tracking using any Defect Management Tool.
* Test Reporting
* Modifying Tests and performing Re & Regression Testing.
* Tracking Defects and Select Test cases for Re & Regression Testing.
* Modifying Test Automation Resources and Maintenance of Resources.

4+ Years

========================

* Selecting or Identifying Areas/Test cases for Automation.
* Designing & Implementing Test Automation Strategy.
* Creating Automation Test Plan and getting approvals.
* Choose selenium tools and Configuring Selenium Test Environment (Ex: Eclipse IDE, Java, Selenium WebDriver and TestNG etc…).
* Involvement in Selenium Environment Setup…
* Automation Framework Design and Implementation.
* Creating, Organizing, and managing Test Automation Resources.
* Creating, Enhancing, debugging and Running Test Cases.
* Organizing, monitoring defect management process.
* Handling changes and conducting Regression Testing.
* Finding solutions for Object Identification issues and error handling issues.
* Co-coordinating Test team members and Development team in order to resolve the issues.
* Interacting with client side people to solve issues and update status.

**17. What are your roles and responsibilities as part of automation in your project?**

**Answer :** Please refer above answer

**18. What are the major challenges in Functional Test Automation?**

**Answer:**

1. Scripting expertise—a high bar for testing talent

2. When you can’t find a trace of tractability

3. Quickly scaling test environments is a challenge

4. Too many UI tests can break testing

5. A lack of transparency can inhibit automated software testing

**19. What are the difficulties you have faced in Object Identification?**

**Answer:**

1. Handling windows based Pop-Up Windows

2. Identifying Dynamic Elements

Many web apps or websites often have web elements that are dynamic in nature, which are not visible when you visit the site for the first time. This means that the web pages are user-specific and display different data for different users based on their requirements; new data appears on the web page after a certain period of time or when a user clicks something on the page. For example, if the ID of an element is changing on every page load, then it’s not easy to handle this situation in a normal way.

3. Timeout or Sync Issue

Whether you call it a timeout or sync issue, it is one of the most common challenges in Selenium test automation. If you don’t handle this issue carefully, most of your testing script might fail. It is even proved many times that around 80% of scripts fail due to improper sync while executing automation testing.

4. Page Loading

As mentioned earlier, some of the web pages in a web app are user-specific and load different elements depending on the user. Some features even appear based on the user’s previous activity. For example, if you have a drop-down menu for Italian food, then food items related to that category will appear in the food dropdown. During the runtime, the Selenium script might not be able to identify the element. Therefore, to overcome this issue, you can use explicit waits to provide elements enough time to load and to discover the element.

**20. How you organized your Test Automation resources in your Project ?**

**Answer:**

1. Plan Your Test Cases & Test Suites

2. Centralize Your Test Assets

3. Differentiate Test Objects

4. Validate & Remove Outdated Test Cases

5. Separate Test Architecture

**21. Did you use any build management tools in your project?**

**Answer:** Yes

Build tools are programs that automate the creation of executable applications from source code. Building incorporates compiling, linking and packaging the code into a usable or executable form. ... Using an automation tool allows the build process to be more consistent.

Some of most popular tools in this category "Java Build Tools" are Gradle, Apache Maven, Sonatype Nexus, Apache Ant, and Pants are the.

**22. How you handled errors in your Test Scripts?**

**Answer:**Please refer below link  
<https://www.automationtestinginsider.com/2019/10/selenium-questions-part5-selenium.html>

Common Exceptions in Selenium

1. **ElementNotVisibleException** - This type of Selenium exception occurs when an existing element in DOM has a feature set as hidden.

2. **ElementNotSelectableException**- This Selenium exception occurs when an element is presented in the DOM, but you can be able to select. Therefore, it is not possible to interact.

3. **NoSuchElementException**- This Exception occurs if an element could not be found.

4. **NoSuchFrameException**- This Exception occurs if the frame target to be switched to does not exist.

5. **NoAlertPresentException**- This Exception occurs when you switch to no presented alert.

6. **NoSuchWindowException**- This Exception occurs if the window target to be switch does not exist.

7. **StaleElementReferenceException**- This Selenium exception occurs when the web element is detached from the current DOM.

8. **SessionNotFoundException**- The WebDriver is acting after you quit the browser.

9. **TimeoutException**- Thrown when there is not enough time for a command to be completed. For Example, the element searched wasn't found in the specified time.

10. **WebDriverException**- This Exception takes place when the WebDriver is acting right after you close the browser.

11. **ConnectionClosedException**- This type of Exception takes place when there is a disconnection in the driver.

12. **ElementNotInteractableException**- This Selenium exception is thrown when any element is presented in the DOM. However, it is impossible to interact with such an element.

13. **InvalidArgumentException**- It occurs when an argument does not belong to the expected type.

14. **SessionNotCreatedException**- It happens when a new session could not be successfully created.

15. **ElementNotVisibleException**- If selenium tries to find an element but the element is not visible within the page

16. **NoSuchAttributeException**- While trying to get attribute value but the attribute is not available in DOM.

**23. Did you create any reusable components while automation any of your projects?**

**Answer:** Yes.  
As an automation test engineer, you might have noticed that some of our test steps get repeated very often in multiple tests.  In such cases, designing the tests in a such a way that it could be reused in multiple workflow is very important. Re-usability allows us to be more efficient & to write better and clean code.

Lets see how we could design reusable tests.

Lets consider a sample application which has below workflows.

* An user can register himself in the application by ordering a product.
* An existing registered user can also order a new product.
* Once a product is ordered, it can be viewed.
* User can logout.

Based on the above requirements, we can come up with different workflows as shown in below.

* A new user enters details -> orders a product -> logout
* A new user enters details -> orders a product -> views product -> logout
* An existing user  -> orders a product -> views product -> logout
* An existing user  -> orders a product -> logout
* An existing user  -> views product -> logout

We might be interested in testing all these above workflows. The same workflow should also be tested for different input parameters. For ex: Different types of products, different payment methods like Credit card, promocodes etc. So writing reusable tests is very important to maintain these tests in future to avoid duplication in your code.

**24. Did you find any test scenarios that cannot be automated in your projects using Selenium?**

**Answer:** Below are some scenarios I think are not beneficial to automate using Selenium.

**CAPTCHA scenarios:** Well, CAPTCHAS are there for purpose. To bypass automation. Best way to handle it is to tell your dev team to disable it or make it static.

Video streaming scenarios: More often that not, Selenium won’t be able to recognise video controls. JavaScript Executor and flex-ui-selenium will work to some extent, but they are not entirely reliable.

**Code reading scenarios:** If your web app has a functionality which reads barcodes or QR codes, it’s not beneficial to automate it. There may be some tools available for them but I’m not sure how effective they are.

**Crash recovery scenarios:** You might want to test your application’s crash recovery. This is a scenario best tested manually. I am not saying you won’t be able to test it using Selenium. You may be. But I don’t know how feasible and beneficial it would be.

Performance testing: It can be automated but it’s best not to automate performance testing using Selenium.

**25. Explain Automation Life Cycle (ATLC) ?**

**Answer:**

**1- Automation feasibility analysis:** The main objective of this phase will be to check feasibility of automation.

So your main focus will be on below points.

* Which test case can be automated and how we can automate them?
* Which module of your application can be tested and which can not be automated
* Which tools we can use for our application (like Selenium,QTP,Sahi,OATS, Telrik etc) and which tools will be best of our application
* Take following factors into consideration like Team size,Effort and cost involved for tools which we will use.

**2- Test Plan/Test Design:**

This phase plays very important role in Automation test life cycle. In this phase you have to create a Test plan by considering below point into considerations.

* Fetch  all the manual test case from test management tool that which TC has to automate.
* Which framework to use and what will be advantage and disadvantage  of the framework which we will use.
* Create a test suite for Automation test case in Test Management tool.
* In test plan you can mention background, limitation, risk and dependency between application and tools.
* Approval from client/ Stack holders.

**3- Environment Setup:** By name itself you can understand that we need to setup machine or remote machine where our test case will execute.

* In this section you can mention how many machine you want.
* What should be the configuration in terms of hardware and software.

**4-Test Script development/ Automation testcase development:**

In this phase you have to start develop automation script and make sure all test script is running fine and should be stable enough.

* Start creating test script based on your requirement
* Create some common method or function that you can reuse throughout your script
* Make your script easy, reusable,well structured and well documented so if third person check your script then he/she can understand your scripts easily.
* Use better reporting so in case of failing you can trace your code
* Finally review your script and your script should be ready before consumption.

**5-Test script execution:**In this phase you have to execute all your test script.

 Some points to remember while execution

* Your script should cover all the functional requirement as per test case.
* Your script should be stable so it should run in multiple environment and multiple browsers (depends on your requirement)
* You can do batch execution also if possible so it will save time and effort.
* In case of failure your script should take screen shots.
* If test case is failing due to functionality, you have to raise a bug/defect

**6- Generate test result / Analyses of  result**:

This is the last phase of Automation test life cycle in which we will gather test result and will share with team/client/stack holders.

* Analyze the output and calculate how much time it take to complete the testcase.
* You should have good report generation like Extent Report ,XSLT report, TestNG report, ReporterNG etc.

**26. Does manual bring more ROI or automation brings more ROI?**

**Answer:** The goal of automated testing is to improve software quality while testing faster and reducing costs, and there is more to the ROI of automation than accounting for manual and regression tests. ... Without proper parallel testing and the coverage it can provide, you risk encountering defects further downstream.

**27. Why did you choose Selenium in your project, when there are so many tools?**

**Answer:**Selenium gained popularity because of one single reason: it is free while other testing tools such as HP QTP are insanely priced. Although QTP is better in terms of easier to learn, better support, and have cool features such as Object Repository, their pricing is unjustifiable.

In an organisation, there would be many projects running and buying licenses for paid automation tool for each projects would burn a very big hole in the company’s wallet.

Second, Selenium is cross platform. You can execute Selenium scripts in Linux and Mac OS where as other tools are tied to Windows Platform. These two are the primary reasons why Selenium is a popular test automation tool.

**28. Criteria for selecting test cases for automation**

**Answer:**  Following are the criteria:

1. Criteria for selecting test cases for automation

2. Tests that use multiple data values for the same actions (data driven tests)

3. Complex and time consuming tests

4. Tests involving many simple, repetitive steps

5. Testing needed on multiple combinations of OS, DBMS & Browsers

6. Test Cases that are very tedious or difficult to perform manually

**29. Which type of test cases exclude for automation**

**Answer:** 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.

**30. Main stages in automation testing life cycle?**  
 **Answer:**Below are the main states in ATLC

* Determining The Scope Of Test Automation
* Selecting The Right Tool For Automation
* Test Plan + Test Design + Test Strategy
* Setting Up The Test Environment
* Automation Test Script Development + Execution
* Analysis + Generation Of Test Reports

**31. What are the main task during planning phase of automation testing?**

**Answer:** During the test planning phase, the testing team decides the test procedure creation standards and guidelines; hardware; software and network to support test environment; a preliminary test schedule; test data requirements; defect tracking procedure and associated tracking tool and a procedure to control test.

**32. Principal features of good automation tool.**

**Answer:**A good automation tool should have the following characteristics.

* Quick and easy test environment setup.
* Cross-platform support.
* Good debugging/logging support.
* Robust object identification.
* Object and image testing abilities.
* Cross browser testing support.
* Database integration and validation.

**33. Different approach for designing automation solution**

**Answer:** These approaches include:

* Keyword Driven Testing
* Page Object Model
* Behavior Driven Development

**34. How to measure success of automation testing?**

**Answer:**  As you look to adopt an automated testing process to meet the rising demand for faster delivery cycles and bug-free releases, it’s vital to assess whether the return on investment (ROI) is worth the change. Before executing, or even thinking of building out an automation strategy, you’ll want to calculate the net gain you’ll see from transitioning. Divide this by the net investment needed to transition (i.e., the tools and resources you use), and you’ll get your ROI for automated testing.

This equation will look like the following:  (Gain- Investment)/Investment

The key question that arises is, “what defines the gains and the investments?” Calculate the 6 measurements outlined in this white paper to estimate the long and short term monetary value you will receive from investing in automation. Before we dive into the benefits of transitioning and the investments in tools and resources you’ll have to make, let’s dive into the common pitfalls in calculating the ROI.

1. Only accounting for creating, developing, and maintaining automated tests versus manual tests. Manual testing will always be important. While automation is on your mind, there are scenarios that will always require manually executed test cases.

2. Not accounting for the percentage of tests that need to stay manual. Redundant or repetitive test steps are great candidates for automation, as having to run multiple of the same test type can be tedious and ultimately prone to human error.

3. Not syncing your automation tool stack with organizational capabilities. To implement an automation strategy, you’ll need to have both automation knowledge and product knowledge.

4. Not accounting for ROI over a period of time. When building out a business case to transition to automation, you’ll not only want to gauge the short-term benefits of investing, but also how it will impact your team and organization in the long run.

**35. If sprint is of 2 weeks, what about automation cycle?**

**Answer:**Answer will be provided soon

**36. How do you estimate and how to track your automation test cases ?**

**Answer:**Answer will be provided soon

**37. How to calculate ROI?**

**Answer:**Please refer answer from question 34

**38. How to calculate automation efforts?**

**Answer:**Please refer below link

<https://www.stepinforum.org/Test_Automation/TA_finaltalk/PDFs/Papers/TEST%20AUTOMATION%20EFFORT%20ESTIMATION.pdf>