**1) Why testing is required?**

Ans: Testing is required for an effective performance of software application or product. It's important to ensure that the application should not result into any failures because it can be very expensive in the future or in the later stages of the development. It's required to stay in the business.

**2) What types of application we test?**

Web based

GUI based/windows based

Web services

Mobile based : native, mobile web apps, hybrid

**3) What is SDLC and different phases in SDLC?**

Ans: SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

Phases in SDLC:

1. Requirement gathering and analysis
2. Design
3. Implementation/ Coding
4. Testing
5. Deployment
6. Maintenance

**1) Requirement gathering and analysis:**  Business requirements are gathered in this phase. This phase is the main focus of the project managers and stake holders. Meetings with managers, stake holders and users are held in order to determine the requirements.

After requirement gathering these requirements are analyzed and then used to plan the basic project approach and to conduct product feasibility study in the economical, operational, and technical areas.

Finally, a Requirement Specification document is created which serves the purpose of guideline for the next phase of the model.

High level requirements will be written in BRD (Business Requirement document) by Business Analyst

**2)  Design:**  In this phase the system and software design is prepared from the requirement specifications which were studied in the first phase. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The system design specifications serve as input for the next phase of the model.

In this phase the testers comes up with the [**Test strategy**](http://istqbexamcertification.com/what-are-the-test-approaches-or-strategies-in-software-testing/), where they mention what to test, how to test.

The define stage Business Analyst define more details about requirements (which are in BRD) in the form of SRS (software requirement specification) or Use Case diagram.

As part of design, Senior Developers write High Level Design Document (HLD)

Developers write Low Level Design Document (LLD)

Seniors Tester write Test Planning document

**3)  Implementation / Coding:**  On receiving system design documents, the work is divided in modules/units and actual coding is started. Since, in this phase the code is produced so it is the main focus for the developer. This is the longest phase of the software development life cycle.

Developers write the code for the requirements.

Testers write test cases as per SRS.

**4)** **Testing**: After the code is developed it is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase. During this phase all types of functional testing like unit testing, integration testing, system testing, acceptance testing are done as well as non-functional testing are also done.

**5) Deployment:** After successful testing the product is delivered / deployed to the customer for their use.

As soon as the product is given to the customers they will first do the beta testing. If any changes are required or if any bugs are caught, then they will report it to the engineering team. Once those changes are made or the bugs are fixed then the final deployment will happen.

**6) Maintenance:** Once when the customers starts using the developed system then the actual problems comes up and needs to be solved from time to time. This process where the care is taken for the developed product is known as maintenance.

**3) What is waterfall in SDLC?**

1. It is also called as “Classic Lifecycle Model”. Linear-Sequential model.
2. This model suggests systematic and sequential approach to software development that begins at requirements and gathering and progress through all the phases of life cycle sequentially
3. Development activities carried out sequentially.
4. Review and approach of each phase outputs.
5. Model does not permit going back and forth.
6. If any defect is found, you can revert to the originating phase and start traversing sequentially all over again.
7. This means that any phase in the development process begins only if the previous phase is complete. In waterfall model phases do not overlap.



* Requirements are very well documented, clear and fixed.
* Product definition is stable.
* Technology is understood and is not dynamic.
* There are no ambiguous requirements.
* Ample resources with required expertise are available to support the product.
* The project is short.

**4) What is the process in agile model?**

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product.

Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing.

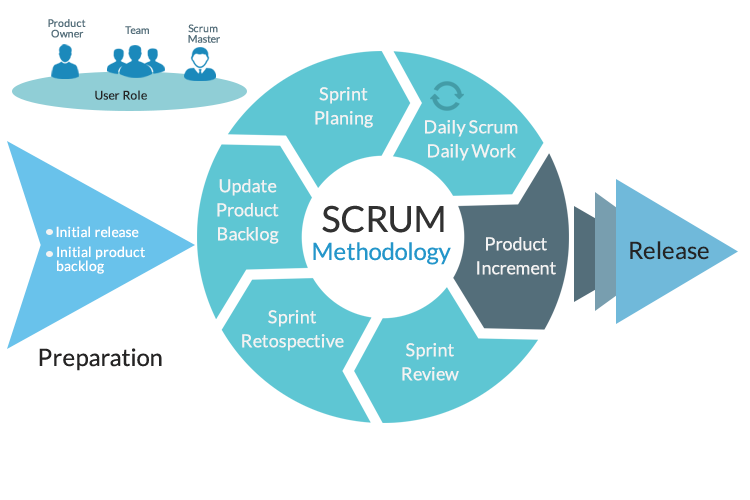
At the end of the iteration a working product is displayed to the customer and important stakeholders.



**5) What is scrum methodology?**

Scrum is an iterative and incremental agile software development framework for managing product development. Scrum is a framework for developing and sustaining complex products. This Guide contains the definition of Scrum. This definition consists of Scrum’s roles, events, artifacts, and the rules that bind them together.

Scrum is not a process or a technique for building products; rather, it is a framework within which you can employ various processes and techniques. Scrum makes clear the relative efficacy of your product management and development practices so that you can improve. The Scrum framework consists of Scrum Teams and their associated roles, events, artifacts, and rules. Each component within the framework serves a specific purpose and is essential to Scrum’s success and usage. The rules of Scrum bind together the events, roles, and artifacts, governing the relationships and interaction between them. The rules of Scrum are described throughout the body of this document.



**6) What is daily standup meeting and what we discuss?**

In [Scrum](https://www.mountaingoatsoftware.com/agile/scrum), on each day of a sprint, the team holds a daily scrum meeting called the "daily scrum.” Meetings are typically held in the same location and at the same time each day. Ideally, a daily scrum meeting is held in the morning, as it helps set the context for the coming day's work. These scrum meetings are strictly time-boxed to 15 minutes. This keeps the discussion brisk but relevant.

[Scrum](https://en.wikipedia.org/wiki/Scrum_(development))-style daily stand-ups involve asking and answering three questions.[[7]](https://en.wikipedia.org/wiki/Stand-up_meeting#cite_note-7) Though it may not be practical to limit all discussion to these three questions, the goal is to stick as closely as possible to these questions:

1. What did I accomplish yesterday?
2. What will I do today?
3. What obstacles are impeding my progress?

By focusing on what each person accomplished yesterday and will accomplish today, the team gains an excellent understanding of what work has been done and what work remains. The daily scrum meeting is not a status update meeting in which a boss is collecting information about who is behind schedule. Rather, it is a meeting in which team members make commitments to each other.

**7) What is user story/feature/sprint back log items and tasks in user story?**

The User Stories are commonly used to describe the product features and will form part of the Scrum Artifacts – Product Backlog and Sprint Backlog.

In Scrum projects, the Product Backlog is a list of user stories. These User Stories are prioritized and taken into the Sprint Backlog in the Sprint Planning Meeting.

Estimation is also based on user stories and the size of the product is estimated in User Story Points.

The sprint backlog is a list of tasks identified by the Scrum team to be completed during the Scrum sprint. During the sprint planning meeting, the team selects some number of product backlog items, usually in the form of user stories, and identifies the tasks necessary to complete each user story.

User stories were on the product backlog and tasks were identified during sprint planning and became part of the sprint backlog.

A user story is typically functionality that will be visible to end users. Developing it will usually involve a programmer and tester, perhaps a user interface designer or analyst, perhaps a database designer, or others.

It would be very rare for a user story to be fully developed by a single person. (And when that did happen, the person would be filling multiple of those roles.)

A task, on the other hand, is typically something like code this, design that, create test data for such-and-such, automate that, and so on. These tend to be things done by one person.

The product backlog can address just about anything, to include new functionality, bugs, and risks. Product backlog items (PBI’s) must be small enough to complete during a sprint and should be small enough to complete within a few days. All stories must be verified that they are implemented to the satisfaction of the Product Owner.

**8) What is sprint planning and spring retro?**

**Sprint Planning:**

Sprint planning is a time boxed working session that lasts roughly 1 hour for every week of a sprint.  In sprint planning, the entire team agrees to complete a set of product backlog items.  This agreement defines the sprint backlog and is based on the team’s velocity or capacity and the length of the sprint.

Sprint planning is a collaborative effort involving a Scrum Master, who facilitates the meeting, a Product Owner, who clarifies the details of the product backlog items and their respective acceptance criteria, and the Entire Agile Team, who define the work and effort necessary to meet their sprint commitment.

**Spring Retro:**

The sprint retrospective is a meeting facilitated by the Scrum Master at which the team discusses the just-concluded sprint and determines what could be changed that might make the next sprint more productive. The sprint review looks at *what* the team is building, whereas the retrospective looks at *how* they are building it.  
  
The retrospective includes three main questions/points for discussion:

* What went well during the sprint cycle?
* What went wrong during the sprint cycle?
* What could we do differently to improve?

The sprint retrospective is an important mechanism that allows a team to continuously evolve and improve throughout the life of a project.  
  
It is important that everyone, including the team, product owner, and Scrum Master, get a chance to air their opinions in an open, honest, yet constructive atmosphere. It often also helps management to get feedback from the team about the work and progress of project.

**9) What is burndown chart and velocity?**

A burn down chart is a graphical representation of work left to do versus time. The outstanding work (or backlog) is often on the vertical axis, with time along the horizontal. That is, it is a run chart of outstanding work. It is useful for predicting when all of the work will be completed. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

Outstanding work can be represented in terms of either time or story points.

The burndown is a chart that shows how quickly you and your team are burning through your customer's user stories. It shows the total effort against the amount of work we deliver each iteration.

**10) What is product backlog item and sprint backlog items?**

**Product Backlog Item:** Product Backlog Items (PBIs) are the elements that make up the Product Backlog. Product Backlog Item Overview:

Product Backlog Items can range from specifications and requirements, to use cases, epics, User Stories, or even bugs and chores. Each PBI must have these qualities:

Description: What the goal of the PBI is.

Value: the Business Value of the PBI as determined by the Product Owner.

Estimate: the Team needs to estimate the relative effort it will take to move the PBI to Done.

Order: The Product Owner needs to prioritize PBIs by their relative value.

**Sprint Back Log Items:** The sprint backlog is a list of tasks identified by the Scrum team to be completed during the Scrum sprint. During the sprint planning meeting, the team selects some number of product backlog items, usually in the form of user stories, and identifies the tasks necessary to complete each user story.

User stories were on the product backlog and tasks were identified during sprint planning and became part of the sprint backlog.

**11) What is user acceptance criteria test cases?**

Acceptance criteria define the boundaries of a user story, and are used to confirm when a story is completed and working as intended.

Acceptance Criteria are the conditions that a software product must satisfy to be accepted by a user, customer, or other in the case of system level functionality, the consuming system.

Acceptance Criteria are a set of statements, each with a clear pass/fail result, that specify both functional and non-functional requirements, and are applicable at the Epic, Feature, and Story Level.

**12) What is v model?**

The V - model is SDLC model where execution of processes happens in a sequential manner in V-shape. It is also known as Verification and Validation model.

V - Model is an extension of the waterfall model and is based on association of a testing phase for each corresponding development stage. This means that for every single phase in the development cycle there is a directly associated testing phase. This is a highly disciplined model and next phase starts only after completion of the previous phase.

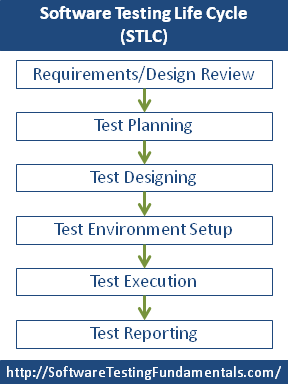
Under V-Model, the corresponding testing phase of the development phase is planned in parallel. So there are Verification phases on one side of the .V. and Validation phases on the other side. Coding phase joins the two sides of the V-Model.



**13) What is STLC?**

Software Testing Life Cycle refers to a testing process which has specific steps to be executed in a definite sequence to ensure that the quality goals have been met. In STLC process, each activity is carried out in a planned and systematic way. Each phase has different goals and deliverables. Different organizations have different phases in STLC; however the basis remains the same.

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **Activity** | **Deliverables** | **Necessity** |
| Requirements/ Design Review | You review the software requirements/ design (Well, if they exist.) | * ‘Review Defect’ Reports | Curiosity |
| Test Planning | Once you have gathered a general idea of what needs to be tested, you ‘plan’ for the tests. | * [Test Plan](http://softwaretestingfundamentals.com/test-plan/) * Test Estimation * Test Schedule | Farsightedness |
| Test Designing | You design/ detail your tests on the basis of detailed requirements/design of the software (sometimes, on the basis of your imagination). | * [Test Cases](http://softwaretestingfundamentals.com/test-case/) / [Test Scripts](http://softwaretestingfundamentals.com/test-script/) /Test Data * Requirements Traceability Matrix | Creativity |
| Test Environment Setup | You setup the test environment (server/ client/ network, etc) with the goal of replicating the end-users’ environment. | * Test Environment | Rich company |
| Test Execution | You execute your Test Cases/ Scripts in the Test Environment to see whether they pass. | * Test Results (Incremental) * [Defect Reports](http://softwaretestingfundamentals.com/defect-report/) | Patience |
| Test Reporting | You prepare various reports for various stakeholders. | * Test Results (Final) * Test/ Defect Metrics * Test Closure Report * Who Worked Late & on Weekends (WWLW) Report [Depending on how fussy your Management is] | Diplomacy |

****

**14) What is defect?**

A Software Defect / Bug is a condition in a software product which does not meet a software requirement or end-user expectations. In other words, a defect is an error in coding or logic that causes a program to malfunction or to produce incorrect/unexpected results.

**15) How to arise a defect and what we specify while logging defect?**

During test execution, you may find that you encounter a defect. You can raise this by clicking on the Defect button.

Defect logging, a process of finding defects in the application under test or product by testing or recording feedback from customers and making new versions of the product that fix the defects or the clients feedback.

Defect tracking is an important process in software engineering as Complex and business critical systems have hundreds of defects. One of the challenging factors is managing, evaluating and prioritizing these defects. The number of defects gets multiplied over a period of time and to effectively manage them, defect tracking system is used to make the job easier.

**16) Defect lifecycle?**

Defect life cycle, also known as Bug Life cycle is the journey of a defect cycle, which a defect goes through during its lifetime. It varies from organization to organization and also from project to project as it is governed by the software testing process and also depends upon the tools used.

* **New:** When a new defect is logged and posted for the first time. It is assigned a status NEW.
* **Assigned:** Once the bug is posted by the tester, the lead of the tester approves the bug and assigns the bug to developer team
* **Open**: The developer starts analyzing and works on the defect fix
* **Fixed**: When developer makes necessary code change and verifies the change, he or she can make bug status as "Fixed."
* **Pending retest**: Once the defect is fixed the developer gives particular code for retesting the code to the tester. Since the testing remains pending from the testers end, the status assigned is "pending request."
* **Retest**: Tester does the retesting of the code at this stage to check whether the defect is fixed by the developer or not and change the status to "Re-test."
* **Verified**: The tester re-tests the bug after it got fixed by the developer. If there is no bug detected in the software, then the bug is fixed and the status assigned is "verified."
* **Reopen**: If the bug persists even after the developer has fixed the bug, the tester changes the status to "reopened". Once again the bug goes through the life cycle.
* **Closed**: If the bug is no longer exits then tester assign the status "Closed."
* **Duplicate**: If the defect is repeated twice or the defect corresponds the same concept of the bug, the status is changed to "duplicate."
* **Rejected**: If the developer feels the defect is not a genuine defect than it changes the defect to "rejected."
* **Deferred**: If the present bug is not of a prime priority and if it is expected to get fixed in the next pro, then status "Deferred" is assigned to such bugs
* **Not a bug:** If it does not affect the functionality of the application then the status assigned to a bug is "Not a bug".



**Different types of testing?**

**18) When do we use regression testing?**

Regression testing is nothing but full or partial selection of already executed test cases which are re-executed to ensure existing functionalities work fine.

This testing is done to make sure that new code changes should not have side effects on the existing functionalities. It ensures that old code still works once the new code changes are done.

Regression Testing is required when there is a

* Change in requirements and code is modified according to the requirement
* New feature is added to the software
* Defect fixing
* Performance issue fix

Software maintenance is an activity which includes enhancements, error corrections, optimization and deletion of existing features. These modifications may cause the system to work incorrectly. Therefore, Regression Testing becomes necessary. Regression Testing can be carried out using following techniques:

**[](http://cdn.guru99.com/images/regressiontestingtypes.png)**

Retest All

* This is one of the methods for regression testing in which all the tests in the existing test bucket or suite should be re-executed. This is very expensive as it requires huge time and resources.

Regression Test Selection

* Instead of re-executing the entire test suite, it is better to select part of test suite to be run
* Test cases selected can be categorized as 1) Reusable Test Cases 2) Obsolete Test Cases.
* Re-usable Test cases can be used in succeeding regression cycles.
* Obsolete Test Cases can't be used in succeeding cycles.

Prioritization of Test Cases

* Prioritize the test cases depending on business impact, critical & frequently used functionalities. Selection of test cases based on priority will greatly reduce the regression test suite.

**19) When do we use integration testing?**

In Integration Testing, individual software modules are integrated logically and tested as a group.

A typical software project consists of multiple software modules, coded by different programmers.  Integration testing focuses on checking data communication amongst these modules.

Hence it is also termed as **'I & T'** (Integration and Testing), **'String Testing'** and sometimes 'Thread Testing'.

**Need of Integration Testing:**

Although each software module is unit tested, defects still exist for various reasons like

* A Module in general is designed by an individual software developer whose understanding and programming logic may differ from other programmers. Integration testing becomes necessary to verify the software modules work in unity
* At the time of module development, there are wide chances of change in requirements by the clients. These new requirements may not be unit tested and hence integration testing becomes necessary.
* Interfaces of the software modules with the database could be erroneous
* External Hardware interfaces, if any, could be erroneous
* Inadequate exception handling could cause issues.

**20) When do we use smoke testing and sanity testing?**

**Smoke Testing:**

Smoke testing is preliminary testing to reveal simple failures severe enough to (for example) reject a prospective software release. A smoke tester will select and run a subset of test cases that cover the most important functionality of a component or system, to ascertain if crucial functions of the software work correctly. When used to determine if a computer program should be subjected to further, more fine-grained testing, a smoke test may be called an intake test.

Smoke Testing is a testing technique that is inspired from hardware testing, which checks for the smoke from the hardware components once the hardware's power is switched on. Similarly in Software testing context, smoke testing refers to testing the basic functionality of the build.

If the Test fails, build is declared as unstable and it is NOT tested anymore until the smoke test of the build passes.

Smoke Testing - Features:

* Identifying the business critical functionalities that a product must satisfy.
* Designing and executing the basic functionalities of the application.
* Ensuring that the smoke test passes each and every build in order to proceed with the testing.
* Smoke Tests enables uncovering obvious errors which saves time and effort of test team.
* Smoke Tests can be manual or automated.

**Sanity testing:**

Sanity testing, a software testing technique performed by the test team for some basic tests. The aim of basic test is to be conducted whenever a new build is received for testing. The terminologies such as Smoke Test or Build Verification Test or Basic Acceptance Test or Sanity Test are interchangeably used, however, each one of them is used under a slightly different scenario.

Sanity test is usually unscripted, helps to identify the dependent missing functionalities. It is used to determine if the section of the application is still working after a minor change.

Sanity testing can be narrow and deep. Sanity test is a narrow regression test that focuses on one or a few areas of functionality.

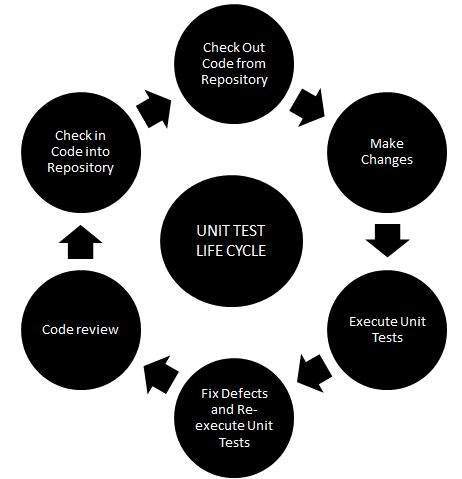
**21) What is unit testing?**

Unit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. It is concerned with functional correctness of the standalone modules.

The main aim is to isolate each unit of the system to identify, analyze and fix the defects.

Unit Testing - Advantages:

* Reduces Defects in the newly developed features or reduces bugs when changing the existing functionality.
* Reduces Cost of Testing as defects are captured in very early phase.
* Improves design and allows better refactoring of code.
* Unit Tests, when integrated with build gives the quality of the build as well.



Unit Testing Techniques:

* Black Box Testing - Using which the user interface, input and output are tested.
* White Box Testing - used to test each one of those functions behavior is tested.
* Gray Box Testing - Used to execute tests, risks and assessment methods.

**22) What is UAT?**

User acceptance testing (UAT) is the last phase of the software testing process. During UAT, actual software users test the software to make sure it can handle required tasks in real-world scenarios, according to specifications.

UAT is one of the final and critical software project procedures that must occur before newly developed software is rolled out to the market.

UAT is also known as beta testing, application testing or end user testing.

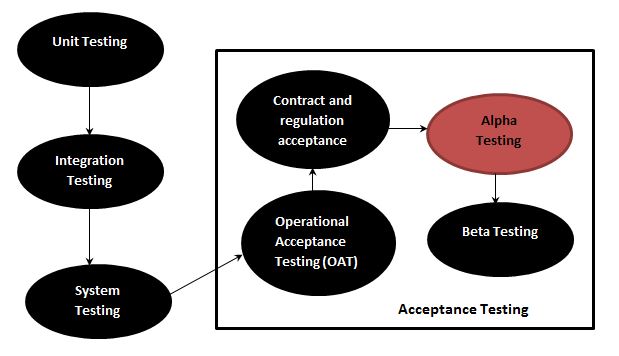
UAT is a process designed to help ensure products will meet user expectations when they are released. It involves running a product through a series of specific tests that help indicate whether or not the product will meet the needs of its users.

**23) What is alpha and beta testing?**

**Alpha testing –** In house virtual user environment can be created for this type of testing. Testing is done at the end of development. Still minor design changes may be made as a result of such testing. Alpha testing takes place at the developer's site by the internal teams, before release to external customers. This testing is performed without the involvement of the development teams.

**Alpha Testing - In SDLC**

The following diagram explains the fitment of Alpha testing in the software development life cycle.

****

In the first phase of alpha testing, the software is tested by in-house developers during which the goal is to catch bugs quickly.

In the second phase of alpha testing, the software is given to the software QA team for additional testing.

Alpha testing is often performed for Commercial off-the-shelf software (COTS) as a form of internal acceptance testing, before the beta testing is performed

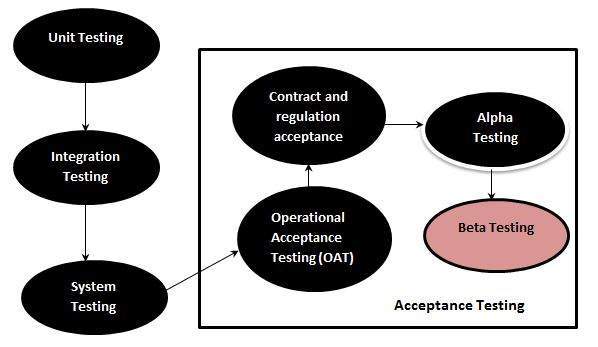
**Beta testing –** Testing typically done by end-users or others. Final testing before releasing application for commercial purpose.

Beta testing also known as user testing takes place at the end users site by the end users to validate the usability, functionality, compatibility, and reliability testing.

Beta testing adds value to the software development life cycle as it allows the "real" customer an opportunity to provide inputs into the design, functionality, and usability of a product. These inputs are not only critical to the success of the product but also an investment into future products when the gathered data is managed effectively.

**Beta Testing - In SDLC**

The following diagram explains the fitment of Beta testing in the software development life cycle:

****

Beta Testing Dependencies

There are number of factors that depends on the success of beta testing:

* Test Cost
* Number of Test Participants
* Shipping
* Duration of Test
* Demographic coverage

**24) When do we use white box testing and block box testing?**

**White box testing:**

White box testing is a testing technique that examines the program structure and derives test data from the program logic/code. The other names of glass box testing are clear box testing, open box testing, logic driven testing or path driven testing or structural testing.

White Box Testing Techniques:

* Statement Coverage - This technique is aimed at exercising all programming statements with minimal tests.
* Branch Coverage - This technique is running a series of tests to ensure that all branches are tested at least once.
* Path Coverage - This technique corresponds to testing all possible paths which means that each statement and branch is covered.

**Advantages of White Box Testing:**

* Forces test developer to reason carefully about implementation.
* Reveals errors in "hidden" code.
* Spots the Dead Code or other issues with respect to best programming practices.

**Disadvantages of White Box Testing:**

* Expensive as one has to spend both time and money to perform white box testing.
* Every possibility that few lines of code are missed accidentally.
* In-depth knowledge about the programming language is necessary to perform white box testing.

**Black box testing:**

Black-box testing is a method of software testing that examines the functionality of an application based on the specifications. It is also known as Specifications based testing. Independent Testing Team usually performs this type of testing during the software testing life cycle.

This method of test can be applied to each and every level of software testing such as unit, integration, system and acceptance testing.

Behavioral Testing Techniques:

There are different techniques involved in Black Box testing.

* Equivalence Class
* Boundary Value Analysis
* Domain Tests
* Orthogonal Arrays
* Decision Tables
* State Models
* Exploratory Testing
* All-pairs testing

**25) What we will do if we don’t have a time to test all stories?**

**26) What we will do if come across any severity issue before release day?**

**27) When do we use automation testing?**

Software Test automation makes use of specialized tools to control the execution of tests and compares the actual results against the expected result. Usually regression tests, which are repetitive actions, are automated.

Testing Tools not only help us to perform regression tests but also helps us to automate data set up generation, product installation, GUI interaction, defect logging, etc.

**Criteria for Tool Selection:**

For automating any application, the following parameters should be considered.

* Data driven capabilities
* Debugging and logging capabilities
* Platform independence
* Extensibility & Customizability
* E-mail Notifications
* Version control friendly
* Support unattended test runs

**Types of Frameworks:**

Typically, there are 4 test automation frameworks that are adopted while automating the applications.

* Data Driven Automation Framework
* Keyword Driven Automation Framework
* Modular Automation Framework
* Hybrid Automation Framework

Popular Tools that are used for Functional automation:

|  |  |  |
| --- | --- | --- |
| Product | Vendor | URL |
| Quick Test Professional | HP | www.hp.com/go/qtp |
| Rational Robot | IBM | http://www-03.ibm.com/software/products/us/en/robot/ |
| Coded UI | Microsoft | http://msdn.microsoft.com/en-us/library/dd286726.aspx |
| Selenium | Open Source | http://docs.seleniumhq.org/ |
| Auto IT | Open Source | http://www.autoitscript.com/site/ |

Popular Tools that are used for Non-Functional automation:

|  |  |  |
| --- | --- | --- |
| Product | Vendor | URL |
| Load Runner | HP | www.hp.com/go/LoadRunner |
| Jmeter | Apache | jmeter.apache.org/ |
| Burp Suite | PortSwigger | http://portswigger.net/burp/ |
| Acunetix | Acunetix | http://www.acunetix.com/ |

**28) What tester will do in each phase of SDLC?**

**29) Difference between load and performance testing?**

**Load testing:**

Load testing is the process of putting demand on a software system or computing device and measuring its response. Load testing is performed to determine a system's behavior under both normal and anticipated peak load conditions. It helps to identify the maximum operating capacity of an application as well as any bottlenecks and determine which element is causing degradation. When the load placed on the system is raised beyond normal usage patterns to test the system's response at unusually high or peak loads, it is known as stress testing. The load is usually so great that error conditions are the expected result, but there is no clear boundary when an activity ceases to be a load test and becomes a stress test.

The term "load testing" is often used synonymously with concurrency testing, software performance testing, reliability testing, and volume testing. All of which are types of non-functional testing that are part of functionality testing used to validate suitability for use of any given software.

The term load testing is used in different ways in the professional software testing community. Load testing generally refers to the practice of modeling the expected usage of a software program by simulating multiple users accessing the program concurrently.[1] As such, this testing is most relevant for multi-user systems; often one built using a client/server model, such as web servers. However, other types of software systems can also be load tested. For example, a word processor or graphics editor can be forced to read an extremely large document; or a financial package can be forced to generate a report based on several years' worth of data. The most accurate load testing simulates actual use, as opposed to testing using theoretical or analytical modeling.

**Performance testing:**

Performance testing, a non-functional testing technique performed to determine the system parameters in terms of responsiveness and stability under various workload. Performance testing measures the quality attributes of the system, such as scalability, reliability and resource usage.

**Performance Testing Techniques:**

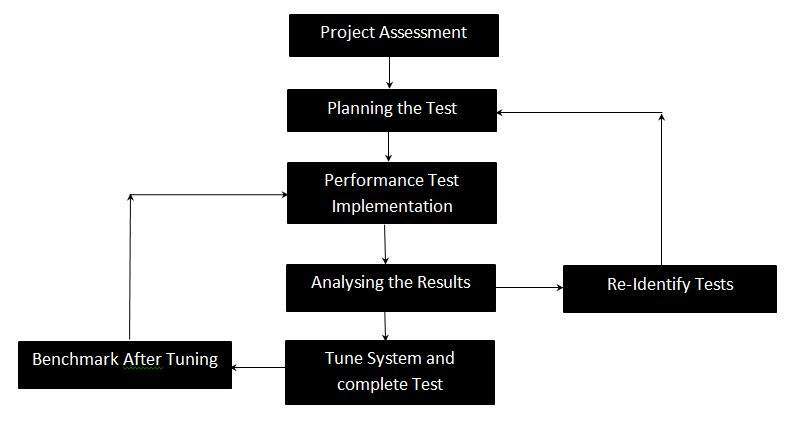
**Load testing** - It is the simplest form of testing conducted to understand the behavior of the system under a specific load. Load testing will result in measuring important business critical transactions and load on the database, application server, etc., are also monitored.

**Stress testing** - It is performed to find the upper limit capacity of the system and also to determine how the system performs if the current load goes well above the expected maximum.

**Soak testing** - Soak Testing also known as endurance testing, is performed to determine the system parameters under continuous expected load. During soak tests the parameters such as memory utilization is monitored to detect memory leaks or other performance issues. The main aim is to discover the system's performance under sustained use.

**Spike testing** - Spike testing is performed by increasing the number of users suddenly by a very large amount and measuring the performance of the system. The main aim is to determine whether the system will be able to sustain the workload.

**Performance Testing Process:**

****

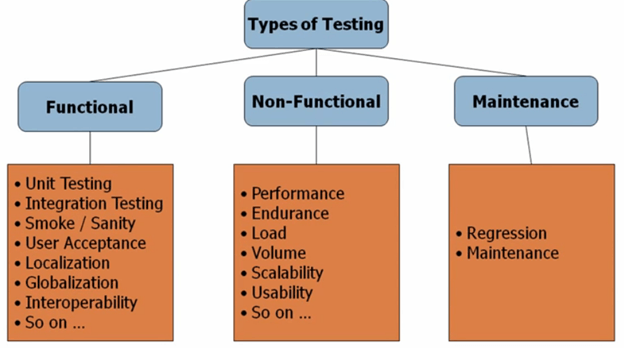
**Attributes of Performance Testing:**

* Speed
* Scalability
* Stability
* Reliability

**Performance Testing Tools**

* Jmeter - http://jmeter.apache.org/
* Open STA - http://opensta.org/
* Load Runner - http://www.hp.com/
* Web Load - http://www.radview.com/

**30) Different types of nonfunctional?**

****

Non-Functional testing is a software testing technique that verifies the attributes of the system such as memory leaks, performance or robustness of the system. Non-Functional testing is performed at all test levels.

**Non-Functional Testing Techniques:**

* Baseline testing
* Compatibility testing
* Compliance testing
* Endurance testing
* Load testing
* Localization testing
* Internationalization testing
* Performance testing
* Recovery testing
* Resilience testing
* Security testing
* Scalability testing
* Stress testing
* Usability testing
* Volume testing

**31) Testing types?**

**32) What is test case?**

A test case is a document, which has a set of test data, preconditions, expected results and post conditions, developed for a particular test scenario in order to verify compliance against a specific requirement.

Test Case acts as the starting point for the test execution, and after applying a set of input values, the application has a definitive outcome and leaves the system at some end point or also known as execution post condition.

**Typical Test Case Parameters:**

* Test Case ID
* Test Scenario
* Test Case Description
* Test Steps
* Prerequisite
* Test Data
* Expected Result
* Test Parameters
* Actual Result
* Environment Information
* Comments

Example:

Let us say that we need to check an input field that can accept maximum of 10 characters.

While developing the test cases for the above scenario, the test cases are documented the following way.

In the below example, the first case is a pass scenario while the second case is a FAIL.

|  |  |  |  |
| --- | --- | --- | --- |
| Scenario | Test Step | Expected Result | Actual Outcome |
| Verify that the input field that can accept maximum of 10 characters | Login to application and key in 10 characters | Application should be able to accept all 10 characters. | Application accepts all 10 characters. |
| Verify that the input field that can accept maximum of 11 characters | Login to application and key in 11 characters | Application should NOT accept all 11 characters. | Application accepts all 10 characters. |

If the expected result doesn't match with the actual result, then we log a defect. The defect goes through the defect life cycle and the testers address the same after fix.

**33) What is test plan/test strategy document?**

Test planning, the most important activity to ensure that there is initially a list of tasks and milestones in a baseline plan to track the progress of the project. It also defines the size of the test effort.

It is the main document often called as master test plan or a project test plan and usually developed during the early phase of the project.

**Test Plan Identifiers:**

|  |  |  |
| --- | --- | --- |
| S.No. | Parameter | Description |
| 1. | Test plan identifier | Unique identifying reference. |
| 2. | Introduction | A brief introduction about the project and to the document. |
| 3. | Test items | A test item is a software item that is the application under test. |
| 4. | Features to be tested | A feature that needs to tested on the testware. |
| 5. | Features not to be tested | Identify the features and the reasons for not including as part of testing. |
| 6. | Approach | Details about the overall approach to testing. |
| 7. | Item pass/fail criteria | Documented whether a software item has passed or failed its test. |
| 8. | Test deliverables | The deliverables that are delivered as part of the testing process,such as test plans, test specifications and test summary reports. |
| 9. | Testing tasks | All tasks for planning and executing the testing. |
| 10. | Environmental needs | Defining the environmental requirements such as hardware, software, OS, network configurations, tools required. |
| 11. | Responsibilities | Lists the roles and responsibilities of the team members. |
| 12. | Staffing and training needs | Captures the actual staffing requirements and any specific skills and training requirements. |
| 13. | Schedule | States the important project delivery dates and key milestones. |
| 14. | Risks and Mitigation | High-level project risks and assumptions and a mitigating plan for each identified risk. |
| 15. | Approvals | Captures all approvers of the document, their titles and the sign off date. |

Test Planning Activities:

* To determine the scope and the risks that need to be tested and that are NOT to be tested.
* Documenting Test Strategy.
* Making sure that the testing activities have been included.
* Deciding Entry and Exit criteria.
* Evaluating the test estimate.
* Planning when and how to test and deciding how the test results will be evaluated, and defining test exit criterion.
* The Test artefacts delivered as part of test execution.
* Defining the management information, including the metrics required and defect resolution and risk issues.
* Ensuring that the test documentation generates repeatable test assets.

The Test Strategy Document is a living document that is created in the project’s Requirements Definition phase, after the Requirements have been specified. The Test Strategy document describes the scope, approach, resources and schedule for the testing activities of the project. This includes defining what will be tested, who will perform testing, how testing will be managed, and the associated risks and contingencies. The Test Strategy document is maintained throughout the life of a project.

**34) Types of testing?**

Ans: **Black box testing** – Internal system design is not considered in this type of testing. Tests are based on requirements and functionality.

**White box testing** – This testing is based on knowledge of the internal logic of an application’s code. Also known as Glass box Testing. Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths, conditions.

**Unit testing** – Testing of individual software components or modules. Typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code may require developing test driver modules or test harnesses.

**Incremental integration testing** – Bottom up approach for testing i.e continuous testing of an application as new functionality is added; Application functionality and modules should be independent enough to test separately done by programmers or by testers.

**Integration testing** – Testing of integrated modules to verify combined functionality after integration. 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.

**Functional testing** – This type of testing ignores the internal parts and focus on the output is as per requirement or not. Black-box type testing geared to functional requirements of an application.

**System testing** – Entire system is tested as per the requirements. Black-box type testing that is based on overall requirements specifications, covers all combined parts of a system.

**End-to-end testing** – Similar to system testing, involves testing of a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate.

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

**Regression testing** – Testing the application as a whole for the modification in any module or functionality. Difficult to cover all the system in regression testing so typically automation tools are used for these testing types.

**Acceptance testing** -Normally this type of testing is done to verify if system meets the customer specified requirements. User or customer do this testing to determine whether to accept application.

**Load testing** – It’s a performance testing to check system behavior under load. Testing an application under heavy loads, such as testing of a web site under a range of loads to determine at what point the system’s response time degrades or fails.

**Stress testing** – System is stressed beyond its specifications to check how and when it fails. Performed under heavy load like putting large number beyond storage capacity, complex database queries, continuous input to system or database load.

**Performance testing** – Term often used interchangeably with ‘stress’ and ‘load’ testing. To check whether system meets performance requirements. Used different performance and load tools to do this.

**Usability testing** – User-friendliness check. Application flow is tested, Can new user understand the application easily, Proper help documented whenever user stuck at any point. Basically system navigation is checked in this testing.

**Install/uninstall testing**– Tested for full, partial, or upgrade install/uninstall processes on different operating systems under different hardware, software environment.

**Recovery testing** – Testing how well a system recovers from crashes, hardware failures, or other catastrophic problems.

**Security testing** – Can system be penetrated by any hacking way. Testing how well the system protects against unauthorized internal or external access. Checked if system, database is safe from external attacks.

**Compatibility testing** – Testing how well software performs in a particular hardware/software/operating system/network environment and different combination s of above.

**Comparison testing** – Comparison of product strengths and weaknesses with previous versions or other similar products.

**Alpha testing** – In house virtual user environment can be created for this type of testing. Testing is done at the end of development. Still minor design changes may be made as a result of such testing.

**Beta testing** – Testing typically done by end-users or others. Final testing before releasing application for commercial purpose.

**35) Exit and Entry criteria?**

**Exit criterion:**

It is used to determine whether a given test activity has been completed or NOT. Exit criteria can be defined for all of the test activities right from planning, specification and execution.

Exit criterion should be part of test plan and decided in the planning stage.

**Examples of Exit Criteria:**

* Verify if all tests planned have been run.
* Verify if the level of requirement coverage has been met.
* Verify if there are NO Critical or high severity defects that are left outstanding.
* Verify if all high risk areas are completely tested.
* Verify if software development activities are completed within the projected cost.
* Verify if software development activities are completed within the projected timelines.

**Entry criteria:**

Entry criterion is used to determine when a given test activity should start. It also includes the beginning of a level of testing, when test design or when test execution is ready to start.

**Examples for Entry Criterion:**

* Verify if the Test environment is available and ready for use.
* Verify if test tools installed in the environment are ready for use.
* Verify if Testable code is available.
* Verify if Test Data is available and validated for correctness of Data.

**36) What is TDD and BDD (cucumber framework)?**

TDD is an iterative development process. Each iteration starts with a set of tests written for a new piece of functionality. These tests are supposed to fail during the start of iteration as there will be no application code corresponding to the tests. In the next phase of the iteration Application code is written with an intention to pass all the tests written earlier in the iteration. Once the application code is ready tests are run.

Any failures in the test run are marked and more Application code is written/re-factored to make these tests pass. Once application code is added/re-factored the tests are run again. This cycle keeps on happening till all the tests pass. Once all the tests pass we can be sure that all the features for which tests were written have been developed.

**Benefits of TDD**

* Unit test proves that the code actually works
* Can drive the design of the program
* Refactoring allow to improve the design of the code
* Low Level regression test suite
* Test first reduce the cost of the bugs

**Drawbacks of TDD**

* Developer can consider it as a waste of time
* The test can be targeted on verification of classes and methods and not on what the code really should do
* Test become part of the maintenance overhead of a project
* Rewrite the test when requirements change

**37) What is priority and severity in defect?**

**Defect Severity:**

In software testing, defect severity can be defined as the degree of impact a defect has on the development or operation of a component application being tested.

Higher effect on the system functionality will lead to the assignment of higher severity to the bug. Quality Assurance engineer usually determines the severity level of defect.

**Defect severity can be categorized into four class**

**Critical**: This defect indicates complete shut-down of the process, nothing can proceed further

**Major**: It is a highly severe defect and collapse the system. However, certain parts of the system remain functional

Medium: It cause some undesirable behavior, but the system is still functional

**Low**: It won't cause any major break-down of the system

**Defect Priority:**

Defect Priority states the order in which a defect should be fixed. Higher the priority the sooner the defect should be resolved.

Defects that leave the software system unusable are given higher priority over defects that cause a small functionality of the software to fail.

**Defect priority can be categorized into three class**

**Low**: The defect is an irritant but repair can be done once the more serious defect have been fixed

**Medium**: During the normal course of the development activities defect should be resolved. It can wait until a new version is created

**High**: The defect must be resolved as soon as possible as it affects the system severely and cannot be used until it is fixed.

**38) How to estimate test cases?**

**39) What is most challenge defect u came across?**

**40) How to deal the production defects?**

**41) How we learn the functionality of system?**

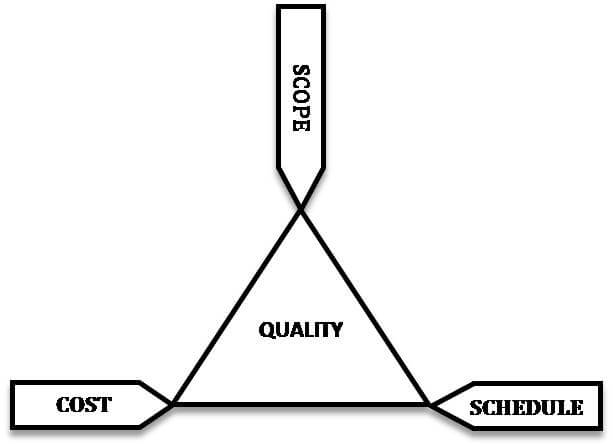
**42) What are the tools to manage defects/stories?**

**43) Who will assign the work?**

**44) Types of test metrics we use normally?**

In software testing, Metric is a quantitative measure of the degree to which a system, system component, or process possesses a given attribute.

In other words, metrics helps estimating the progress, quality and health of a software testing effort.



Software testing metrics - Improves the efficiency and effectiveness of a software testing process."

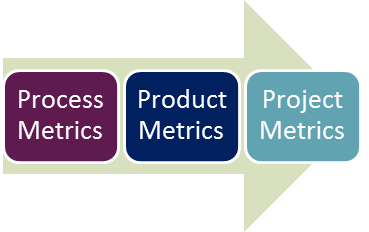
Software testing metrics or software test measurement is the quantitative indication of extent, capacity, dimension, amount or size of some attribute of a process or product.

**Types:**

* **Process Metrics:** It can be used to improve the process efficiency of the SDLC ( Software Development Life Cycle)
* **Product Metrics:** It deals with the quality of the software product
* **Project Metrics:** It can be used to measure the efficiency of a project team or any tools being used by the team members

Identification of correct testing metrics is very important. Few things need to be considered before identifying the test metrics

* Fix the target audience for the metric preparation
* Define the goal for metrics
* Introduce all the relevant metrics based on project needs
* Analyze the cost benefits aspect of each metrics and the project lifestyle phase in which it results into the maximum output.



**45) What is traceability matrix?**

A traceability matrix is a document that co-relates any two-baseline documents that require a many-to-many relationship to check the completeness of the relationship. It is used to track the requirements and to check the current project requirements are met.

**46) What are typical environments we have in projects?**

**47) What is development environment?**

**48) What is QA environment?**

**49) What is production environment?**

**50) What are different defect metrics and measurements we prepare?**

**51) What are weakness and strong points?**

**52) What is staging environment?**

**Disadvantages of Manual Testing:**

* It takes more time or resources
* Less accuracy
* Comparison of data (large amount of data) is impractical
* Performance testing is impractical
* Batch testing takes more time
* Data driven testing is possible but time taking

**Advantages of Test Automation:**

* Fast
* Reliability
* Reusable : Sanity test, Regression testing
* Repeatable : Data driven testing
* Programmable: variables, operators, flow control statements, functions etch.
* Comprehensive : Batch testing

**Disadvantages of Test Automation:**

* Takes more efforts at initial stage
* It is 100% impractical
* Lack of knowledge
* Usability testing is impractical
* Debugging issue
* Not suitable for dynamically changing environments
* Not suitable for dynamically changing UI designs
* Environmental issues
* Tools may have their own defects

**Types of Test Tools:**

**Business classification:**

* **Vendor(Tool developer) tools**
* HP(Take over from Mercury interactive in 2007) Tools –
  + - * Winrunner (scrapped)
      * Loadrunner

**For Performance testing**

* + - * Astra quicktest (Retired)
      * QTP (Retired)
      * UFT (Advanced version of QTP) UFT = QTP + Service tool

**For Functional and Regression test automation**

* + - * TestDirector(Retired)
      * Quality center (Advanced version of TestDirector)
      * ALM (Advanced version of QC)

**For Project Life Cycle Management includes test management (Test management includes defect management)**

* **IBM :** RFT, RPT etc
* **Microfocus**
* **Silk performer**
* **Silk Test etc.**
* **Open Source Tools:** Selenium, Jmeter, Bugzilla etc.
* **In house tools:**

These are private tools. Companies develop tools for internal use.

**Technical Classification:**

* Functional and Regression Test tools

UFT, RFT, Selenium, SilkTest, TestComplete etc.

* Performance Test tools:

LoadRunner, RPT, SilkPerformer, JMeter etc.

* Project Life Cycle Management :

ALM

* Defect Management:

Bugzilla, Issue Tracker, PR Tracker (problem report)

**Overview of Functional Testing:**

Functional Testing: Test type

System Testing: Test level

System Testing Level: Functional and Non-Functional test types

Functional and Unit testing

Functional and Integration testing

Functional and System testing

Functional and Acceptance testing

Covers of Functional Testing:

* Input domain coverage:

Data factors

Type of data

Size of data

Range of data

* Output domain coverage:
* Database testing:

Da manipulations (Add, Update, and Delete operations)

Data integrity

Data retrievals

Data comparisons

Data backup and recovery (System admin) etc.

* Error handling:
* Order of functionalities

**QTP – Quick Test Professional**

QTP is an automated functional testing tool that helps testers to perform automated regression testing in order to identify any gaps, errors/defects in contrary to the actual/desired results of the application under test.

QTP stands for **Q**uick **T**est **P**rofessional, a product of **H**ewlett **P**ackard **(HP)**. This tool helps testers to perform an automated functional testing seamlessly without monitoring once script development is complete.

HP QTP uses **Visual Basic Scripting (VBScript)** for automating the applications. The Scripting Engine need not be installed exclusively as it is available part of the Windows OS. The Current version of VBScript is 5.8 which is available as part of Win 7. VBScript is NOT an object oriented language but an object based language.

**Testing Tools:**

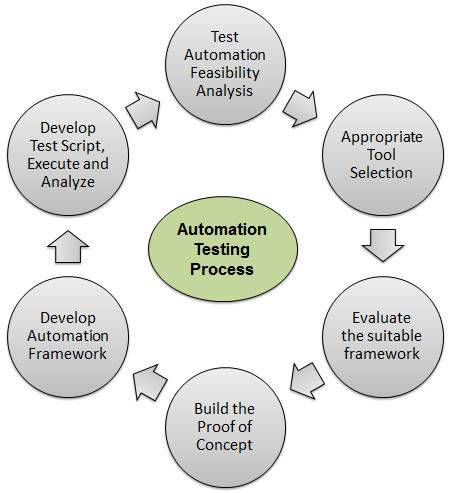
Tools from a software testing context, can be defined as a product that supports one or more test activities right from planning, requirements, creating a build, test execution, defect logging and test analysis.

**Where QTP Fits in?**

QTP is a Functional testing tool which is best suited for regression testing of the applications. QTP is a licensed/commercial tool owned by HP which is one of the most popular tools available in the market. It compares the actual and expected result and reports the results in the execution summary

**Automated Testing Process:**

For any automated tool implementation, the following are the phases/stages of it. Each one of the stages corresponds to a particular activity and each phase has a definite outcome.



* **Test Automation Feasibility Analysis** - First step is to check if the application can be automated or not. Not all applications can be automated due to its limitations.
* **Appropriate Tool Selection** - The Next most important step is the selection of tools. It depends on the technology in which the application is built, its features and usage.
* **Evaluate the suitable framework** - Upon selecting the tool the next activity is to select a suitable framework. There are various kinds of frameworks and each framework has its own significance. We will deal with frameworks in detail later this chapter.
* **Build the Proof of Concept** - Proof of Concept (POC) is developed with an end to end scenario to evaluate if the tool can support the automation of the application. As it is performed with an end to end scenario which will ensure that the major functionalities can be automated.
* **Develop Automation Framework** - After building the POC, framework development is carried out, which is a crucial step for the success of any test automation project. Framework should be built after diligent analysis of the technology used by the application and also its key features.
* **Develop Test Script, Execute and Analyze** - Once Script development is completed, the scripts are executed, results are analyzed and defects are logged, if any. The Test Scripts are usually version controlled.

**Types of Object Repository:**

Object Repository is a collection of object and properties with which QTP will be able to recognize the objects and act on it. When a user records a test, the objects and its properties are captured by default. Without understanding objects and its properties, QTP will NOT be able to play back the scripts.

|  |  |
| --- | --- |
| **Topic** | **Description** |
| [**Object Spy and its Features**](https://www.tutorialspoint.com/qtp/qtp_object_spy.htm) | To Understand the usage of object Spy and its associated functionalities. |
| [**Working with Object Repository**](https://www.tutorialspoint.com/qtp/qtp_add_objects_to_OR.htm) | Adding, Editing, Deleting Objects from an Object Repository and its associated functionalities. |
| [**Types of Object Repository**](https://www.tutorialspoint.com/qtp/qtp_or_types.htm) | Deals with Shared Object and Local Object Repository and their context with respect to scripting. |
| [**User-defined Objects**](https://www.tutorialspoint.com/qtp/qtp_user_defined_objects.htm) | Deals with the circumstances to use the User-Defined Objects. |
| [**Object Repository in XML**](https://www.tutorialspoint.com/qtp/qtp_or_as_xml.htm) | Deals with converting OR's to XML and how to use the object Repository as XML. |
| [**Comparing and Merging OR**](https://www.tutorialspoint.com/qtp/qtp_comparing_and_merging_or.htm) | Operations such as Compare OR', Merge OR's to effectively work with Object Repository. |
| [**Ordinal Identifiers**](https://www.tutorialspoint.com/qtp/qtp_ordinal_identifier.htm) | Circumstances when the ordinal identifiers are used and their advantages. |
| [**Child Objects**](https://www.tutorialspoint.com/qtp/qtp_child_objects.htm) | Using Child Objects for effective scripting |

**Actions:**

Actions helps testers to divide scripts into groups of QTP statements called actions. Actions are similar to functions in VBScript, however there are few differences. By Default QTP creates a test with 1 action.

|  |  |
| --- | --- |
| **Actions** | **Functions** |
| Actions are inbuilt feature of QTP. | VBScript Functions are supported by both VBScript and QTP. |
| Actions parameters are passed by value only. | Function parameters are passed either by by value or by ref. |
| Actions have extension .mts | Functions are saved as .vbs or .qfl |
| Actions may or may not be reusable. | Functions are always reusable. |

The properties of the action can be accessed by Right Clicking on the Script Editor Window and Selecting "Properties".

Action properties contains following information:

* Action Name
* Location
* Reusable Flag
* Input Parameters
* Output Parameters

**Types of Actions:**

There are three types of actions:

* **Non-reusable action** - An action that can be called only in that specific test in which it has been designed and can be called only once
* **Reusable action** - An action that can be called multiple times any test in which it resides and can also be used by any other tests
* **External Reusable action** - It is a reusable action stored in another test. External actions are read-only in the calling test, but it can be used locally with the editable copy of the Data Table information for the external action

**Working with Actions:**

There are three options to insert an action. Click on each one of those to know more about the selected type of action.

|  |  |
| --- | --- |
| **Action Type** | **Description** |
| [**Insert Call to New Action**](https://www.tutorialspoint.com/qtp/qtp_call_to_action.htm) | Inserts a New Action from the existing action |
| [**Insert Call to Copy of Action**](https://www.tutorialspoint.com/qtp/qtp_call_to_copy_of_action.htm) | Inserts a copy of an existing action |
| [**Insert Call to Existing Action**](https://www.tutorialspoint.com/qtp/qtp_call_to_existing_action.htm) | Inserts a call to existing re-usable action |

**Datatables:**

A DataTable, similar to Microsoft Excel helps testers to create data driven test cases that can be used to run an Action multiple times. There are two types of Datatables.

* **Local Data Table** - Each action has its own private data table also known as local data table which is can also be accessed across actions.
* **Global Data Table** - Each test has one global data sheet that is accessible across actions.

**Data Table Operations:**

There are three types of objects to access Data Table. Data Table Operations can be well understood by traversing through the below link:

|  |  |
| --- | --- |
| **Object Type** | **Description** |
| [**Data Table Methods**](https://www.tutorialspoint.com/qtp/qtp_data_table_methods.htm) | Gives Detailed information about the data table methods. |
| [**DTParameter Object Methods**](https://www.tutorialspoint.com/qtp/qtp_dt_parameter.htm) | Gives Detailed information about the DTParameter methods. |
| [**DTSheet Object Methods**](https://www.tutorialspoint.com/qtp/qtp_dt_sheet.htm) | Gives Detailed information about the DTSheet methods. |

**What are CheckPoints?**

Checkpoints, as the name says it all, it refers to a validation point that compares the current value for specified properties or current state of an object with the expected value which can be inserted at any point of time in the script.

**Types:**

|  |  |
| --- | --- |
| **Type** | **Description** |
| Standard Checkpoint | Verifies the property values of an object in application under test and supported by all add-in environments. |
| Bitmap Checkpoint | Verifies an area of your application as a bitmap |
| File Content Checkpoint | Verifies the text in a dynamically generated or accessed file such as .txt,.pdf |
| Table Checkpoint | Verifies the information within a table. Not all environments are supported. |
| Text Checkpoint | Verify if the text that is displayed within a defined area in a Windows-based application, according to specified criteria. |
| Text Area Checkpoint | Verifies if the text string is displayed within a defined area in a Windows-based application, according to specified criteria. |
| Accessibility Checkpoint | Verifies the page and reports the areas of the Web site that may not conform to the World Wide Web Consortium (W3C) Web Content Accessibility Guidelines |
| Page Checkpoint | Verifies the characteristics of a Web page. It can also check for broken links. |
| Database Checkpoint | Verifies the contents of a database accessed by the application under test. |
| XML Checkpoint | Verifies the content of the .xml documents or .xml documents in Web pages and frames. |

**Inserting CheckPoint:**

When the user wants to insert a checkpoint, one has to ensure that most of the checkpoints are supported during the recording sessions only. Once the user stops recording, checkpoints are NOT enabled.

**What is Synchronization?**

Synchronization point is the time interface between Tool and Application under test. Synchronization point is a feature to specify delay time between one step and another of the test script.

For Example, clicking on a link may load the page is 1 second, sometimes 5 seconds or even it might take 10 seconds to load it completely. It depends on various factors such as the application server response time, network bandwidth , client system capabilities etc.

If the time is varying then the script will fail unless the tester handles these time differences intelligently.

Ways to Insert Sync Point:

* WaitProperty
* Exist
* Wait
* Sync(only for web based apps)
* Inserting QTP Inbuilt Synchronization points.

Sometimes, QTP is unable to find any object that matches the recognized object description or it may find more than one object that fits the description, then QTP ignores the recognized description and uses the Smart Identification mechanism to recognize the object.

QTP's Smart Identification uses two types of properties:

**Base Filter Properties** - The basic properties of a particular test object class whose values cannot be changed without changing the essence of the original object.

**Optional Filter Properties** - Other properties also assist in identifying the objects of a particular class whose properties are unlikely to change often but can be ignored if they are no longer applicable.

**Enabling Smart identification for an object:**

**Step 1** : Navigate to "Tools" → "Object Identification". Object Identification Dialog Opens.

**Step 2** : Choose the Environment, Object Class and Turn ON "Enable Smart Identification" as shown below:

**Debugging:**

Debugging, in automation testing context, is a systematic process of spotting and fixing the coding issues in the automation scripts so that the script will be more robust and can spot the defects in the application.

There are various ways to perform debugging using break points in QTP. Break Points can be inserted just by pressing "F9" or by using the Menu option "Run" → "Inserting/Removing Break Point".

After Inserting the Break point the "Red Colored" Dot and the line will be highlighted in RED.

|  |  |  |
| --- | --- | --- |
| **Method** | **ShortCut** | **Description** |
| Step Into | F11 | Used to execute each and every Step. Steps into the Function/Action and executes line by line. It pauses on each line after execution. |
| Step Over | F10 | Used to Step over the Function. Step Over runs only the current step in the active document. |
| Step Out | Shift+F11 | After Step Into the function, you can use the Step Out command. Step Out continues the run to the end of the function and then pauses the run session at the next line. |

**Options in Break Point:**

Various Options in Break Point can be accessed by Navigating 'Run' Menu.

|  |  |
| --- | --- |
| **ShortCut** | **Description** |
| F9 | Insert/Remove BreakPoint |
| Ctrl+F9 | Enable/Disable BreakPoint |
| Ctrl+Shift+F9 | Clear All BreakPoint |
| Use Only Menu | Enable/Disable All BreakPoints |

**Debugging Pane:**

The Following are the panes in the debugging window:

* Debugging using Break Point
* **Output** - This Tab displays all the Output of the Print Statements.
* **Watch** - This Tab displays the Boolean output of the Given Expression.
* **Local Variables** - This Tab displays the Output of the Local Variables.

**Error Handling:**

There are various ways on handling errors in QTP. There are three possible kinds of error type one would encounter while working with QTP.

* Syntax Errors
* Logical Errors
* Run Time Errors

**Error Types:**

**Syntax Errors:**

Syntax errors are the typos or a piece of the code that does not confirm with the VBscripting language grammar. Syntax errors occur at the time of compilation of code and cannot be executed until the errors are fixed. To verify the syntax one use the keyboard shortcut as Ctrl+F7 and the result is displayed as shown below. If the window is NOT displayed one can navigate to "View" → "Errors".

**Logical Errors:**

If the script is syntactically correct but it produces unexpected results. Logical error usually does not interrupt the execution but produces incorrect results. Logical errors could occur due to variety of reasons, viz- wrong assumptions or misunderstanding of the requirement and sometimes incorrect program logics (using do-while instead of do-Until) or Infinite Loops.

One of the ways to detect a logical error is to perform peer reviews and also verifying the QTP output file/result file to ensure the tool has performed what it has intended to do.

**RunTime Errors:**

As The name states, this kind of Error happens during Run Time. The reason for such kind of errors is that the script trying to perform something but it is unable to do so and the script usually stops as it is unable to continue with the execution. Classic Examples for Run Time Errors are,

* File NOT found but the script trying to read the file.
* Object NOT found but script is trying to act on that particular object.
* Dividing a number by Zero.
* Array Index out of bounds while accessing array elements.

**Handling Run-Time Errors:**

There are various ways to handle errors in the code.

**1. Using Test Settings** - Error handling can be defined the Test Settings by Navigating to "File" >> "Settings" >> "Run" Tab as shown below. We can select any of the specified settings and click "OK".

**2. Using On Error Statement** - On Error statement is used to notify the VBScript engine of intentions to handle the run-time errors by tester, rather than allowing the VBScript engine to display error messages that are not user friendly.

* On Error Resume Next - On Error Resume Next informs the VBScript engine to process executing the next line of code when an error is encountered.
* On error Goto 0 - This helps the testers to turn off the error handling.

**3. Using Err Object** - Error object is an inbuilt object within VBScript that captures the run time error number and error description with which we will be able to debug the code easily.

* Err.Number - The Number property returns or Sets a numeric value specifying an error. If Err.Number value is 0 then No error had occured.
* Err.Description - The Description property returns or sets a brief description about an error.
* Err.Clear - The Clear method resets the Err object and clears all the previous values associated with it.

**Using Exit Statement** - Exit Statements can be used along with Err object to exit from a test or action or iteration based on the Err.Number value. Let us see each one of those Exit statements in detail.

* ExitTest - Exits from the entire QTP test no matter what the run-time iteration settings are.
* ExitAction - Exits the current action.
* ExitActionIteration - Exits the current iteration of the action.
* ExitTestIteration - Exits the current iteration of the QTP test and proceeds to the next iteration.

**5. Recovery Scenarios** - Upon encountering an error, recovery scenarios are triggered based on certain conditions.

**6. Reporter Object** - Reporter Object helps us to report an event to the run results. It helps us to identify if the concerned action/step is pass/fail.

**UFT:**

**Version history:**

1. Astra Quicktest (1.0 - 5.0)
2. QTP

5.6- 2002

6.5 – 2003

8.0 – 2004

8.2 – 2005

9.0 – 2006

9.2 – 2007 ---- Mercury interactive

9.5 - 2008 ---- HP

10.0 – 2009

11.0 – 2010

UFT 11.5 – 2012

12.02 – 2014

**UFT product information:**

1. UFT is a functional and regression test tool. Supports desktop and web applications. Selenium is also a functional and regression tool. But it only supports web applications.
2. UFT developed in a.net technology. Whereas Selenium developed in Java technology.
3. UFT is an object based test tool i.e., based on front end objects only UFT supports test operations. LoadRunner is protocol based test tool.
4. UFT has an integrated MS Access database engine to support database operations.
5. UFT has VBScript engine to apply program logic to tests.

Work with flat files

Work with excel files

Work with database files.

**UFT supporting environments:**

1. OS environment
2. MS Windows
3. UNIX

**Application environment:**

1. CUI(Command User Interface) based
2. Windows / Desktop
3. Web based

Selenium – web based applications only

**Scripting / Programming:**

* UFT- VBScript (Visual basic scripting)
* Selenium – Java, PHP, Ruby, Python, C#, Perl

**UFT add in:**

Internal add in:

* Standard Windows (built add in)
* Activex
* Visual basic
* Web

External Add in:

* Java
* .Net for Windows forms
* .Net for Web forms
* WPF
* SAP for GUI
* SAP for Web
* PeopleSoft
* Siebel
* Power builder
* Delphi
* Web services
* TE (Terminal Emulators) For mainframes
* Oracle
* Smalltalk
* Stringray

**How to select appropriate add in’s for AUT (Application Under Test):**

3 Web applications developed using Java technology:

**UFT License:**

1. Seat / Node locked
2. Concurrent / Float

**UFT IDE (Integrated Development Environment):**

3 types of features support:

1. Direct features(Features available in UFT main window):

Recording, Checkpoints, Output values, Data table, Object spy, Step generator, Object depositary etc.

1. Integrated tools from HP:

Password encoder, Test batch runner, Test result deletion tool etc.

1. Integrated tools from others:

Eg: MS access database for database operations, VBScript engine for programming logic, MS script debugger for debugging tests.

**Limitations of UFT:**

* It doesn’t support UNIX or UNIX flavors like UNIX/Linux/Solaris operating environments.
* It only supports IE for creating tests.
* It only supports VBScript for enhancing tests.
* Challenges in object identification

**UFT Test Process:**

**QTP:**

* QTP is Quick Test Professional. It is a flagship functional tool from Mercury interactive now acquired by HP.
* It is an **icon based tool,** which automates the functional and regression testing of an application.
* Scripting language is **VBScript** which is easy to use, understand and program.
* Uses **Active Screen** Technology to record scripts which aids the tester in referring to the screens object properties.
* Library files contain **VBScript functions and subroutines** that can be added to the test.
* QTP enables **Business Process Testing (working in QC)** for non-technical subject matter experts to collaborate effectively with Automation Engineers.
* **Versions of QTP 8.2, 9.5, 10**
* Supports major development environments

**Environments support:**

* Oracle
* Java
* Delphi
* SAP
* A.Net
* ActiveX
* People soft
* Power Builder
* Siebel
* Stingray
* Terminal Emulator
* Visual Basic
* Visual Age
* Web
* Web Services

**Manual Testing:**

|  |
| --- |
| **1. What is baseline testing?**  Baseline testing is the process of running a set of tests to capture performance information. Baseline testing use the information collected to made the changes in the application to improve performance and capabilities of the application. Baseline compares present performance of application with its own previous performance. |
| **2. What is benchmark testing?**  Benchmarking testing is the process of comparing application performance with respect to industry standard which is given by some other organization. Benchmark informs us where our application stands with respect to others. Benchmark compares our application performance with other company’s application’s performance. |
| **3. What is verification and validation?**  **Verification:** process of evaluating work-products of a development phase to determine whether they meet the specified requirements for that phase.  **Validation:** process of evaluating software during or at the end of the development process to determine whether it specified requirements.  **Difference between Verification and Validation:**  - Verification is Static Testing whereas Validations is Dynamic Testing. - Verification takes place before validation. - Verification evaluates plans, document, requirements and specification, whereas Validation evaluates product. - Verification inputs are checklist, issues list, walkthroughs and inspection, whereas in Validation testing of actual product. - Verification output is set of document, plans, specification and requirement documents where as in Validation actual product is output. |
| **4. Explain Branch Coverage and Decision Coverage.**  - Branch Coverage is testing performed in order to ensure that every branch of the software is executed atleast. To perform the Branch coverage testing we take the help of the Control Flow Graph.   - Decision coverage testing ensures that every decision taking statement is executed atleast once.  - Both decision and branch coverage testing is done to ensure the tester that no branch and decision taking statement, will not lead to failure of the software.  - To Calculate Branch Coverage: **Branch Coverage = Tested Decision Outcomes / Total Decision Outcomes.** |
| **5.The differences between Retesting and Regression testing are below:**  - Retesting is done to verify defect fix previous in now working correctly whereas regression is perform to check if the defect fix have not impacted other functionality that was working fine before doing changes in the code.  - Retesting is specific and is performed on the bug which is fixed where as in regression is not be always specific to any defect fix it is performed when any bug is fixed.  - Retesting concern with executing those test cases that are failed earlier whereas regression concern with executing test cases that was passed in earlier builds.  - Retesting has higher priority over regression. |
| **6. What is Mutation testing & when can it be done?**  Mutation testing is a performed to find out the defect in the program. It is performed to find put bugs in specific module or component of the application. Mutation testing is based on two assumptions:  **Competent programmer hypothesis:** according this hypothesis we suppose that program write the correct code of the program. Coupling effect: according to this effect collection of different set of test data can also find large and complex bugs. In this testing we insert few bugs into program to examine the optimal test inputs. |
| **7. What is severity and priority of bug? Give some example.**  **Priority:** concern with application from the business point of view.  It answers: How quickly we need to fix the bug? Or how soon the bug should get fixed? Severity: concern with functionality of application.   **8. How much the bug is affecting the functionality of the application?**  Ex.  1. High Priority and Low Severity: If a company logo is not properly displayed on their website.  2. High Priority and High Severity: Suppose you are doing online shopping and filled payment information, but after submitting the form, you get a message like "Order has been cancelled."  3. Low Priority and High Severity: If we have a typical scenario in which the application get crashed, but that scenario exists rarely.  4. Low Priority and Low Severity: There is a mistake like "You have registered success" instead of successfully, success is written. |
| **9. Explain bug leakage and bug release.**  **Bug Leakage:** When customer or end user discovered a bug which can be detected by the testing team. Or when a bug is detected which can be detected in pervious build then this is called as Bug Leakage.  **Bug release:** is when a build is handed to testing team with knowing that defect is present in the release. The priority and severity of bug is low. It is done when customer want the application on the time. Customer can tolerate the bug in the released then the delay in getting the application and the cost involved in removing that bug. These bugs are mentioned in the Release Notes handed to client for the future improvement chances. |
| **10. What is alpha and beta testing?**  **Alpha testing:** is performed by the IN-House developers. After alpha testing the software is handed over to software QA team, for additional testing in an environment that is similar to the client environment.   **Beta testing:** beta testing becomes active. It is performed by end user. So that they can make sure that the product is bug free or working as per the requirement. IN-house developers and software QA team perform alpha testing. The public, a few select prospective customers or the general public performs beta testing. |
| **11. What is Monkey testing?**  Monkey testing is a type of Black Box Testing used mostly at the Unit Level. In this tester enter the data in any format and check the software is not crashing. In this testing we use Smart monkey and Dumb monkey.  Smart monkeys are used for load and stress testing, they will help in finding the bugs. They are very expensive to develop.  Dumb monkey, are important for basic testing. They help in finding those bugs which are having high severity. Dumb monkey are less expensive as compare to Smart monkeys.  Example: In phone number filed Symbols are entered. |
| **12. What is test driver and test stub?**  - The Stub is called from the software component to be tested. It is used in top down approach. - The driver calls a component to be tested. It is used in bottom up approach. - Both test stub and test driver are dummy software components.  **We need test stub and test driver because of following reason:**  - Suppose we want to test the interface between modules A and B and we have developed only module A. So we cannot test module A, but if a dummy module is prepare, using that we can test module A.  - Now module B cannot send or receive data from module A directly so, in these cases we have to transfer data from one module to another module by some external features. This external feature used is called Driver. |
| **13. What is random testing?**  When tester performs testing of application by using random input from the input domain of the system, this is Random Testing.   **Random testing involve following procedures:**  - Selection of input domain. - Randomly selecting any input from input domain. - Using these test input testing of application is performed. - The results are compared to the system specification. The test is a failure if any input leads to incorrect results, otherwise it is a success. |
| **14. What is Agile Testing?**  Agile Testing means to quickly validation of the client requirements and make the application of good quality user interface. When the build is released to the testing team, testing of the application is started to find the bugs. As a Tester, we need to focus on the customer or end user requirements. We put the efforts to deliver the quality product in spite of short time frame which will further help in reducing the cost of development and test feedbacks will be implemented in the code which will avoid the defects coming from the end user. |
| **15. Describe Use Case Testing.**  **Use Case:** A use case is a description of the process which is performed by the end user for a particular task. Use case contains a sequence of step which is performed by the end user to complete a specific task or a step by step process that describe how the application and end user interact with each other. Use case is written by the user point of view.  **Use case Testing:** the use case testing uses this use case to evaluate the application. So that, the tester can examines all the functionalities of the application. Use case testing cover whole application, |
| **16. What is the purpose of test strategy?**  We need Test Strategy for the following reasons:  1. To have a signed, sealed, and delivered document, where the document contains details about the testing methodology, test plan, and test cases. 2. Test strategy document tells us how the software product will be tested. 3. Test strategy document helps to review the test plan with the project team members. 4. It describes the roles, responsibilities and the resources required for the test and schedule.  5. When we create a test strategy document, we have to put into writing any testing issues requiring resolution.  The test strategy is decided first, before lower level decisions are made on the test plan, test design, and other testing issues. |
| **17. Explain bug life cycle.**  Bug Life Cycle:  - When a tester finds a bug .The bug is assigned with NEW or OPEN status,  - The bug is assigned to development project manager who will analyze the bug .He will check whether it is a valid defect. If not valid bug is rejected then status is REJECTED.  - If not, next the defect is checked whether it is in scope. When bug is not part of the current release .Such defects are POSTPONED  - Now, Tester checks whether a similar defect was raised earlier. If yes defect is assigned a status DUPLICATE  - When bug is assigned to developer. During this stage bug is assigned a status IN-PROGRESS  - Once code is fixed. Defect is assigned a status FIXED  - Next the tester will re-test the code. In case the test case passes the defect is CLOSED  - If the test case fails again the bug is RE-OPENED and assigned to the developer. That’s all to Bug Life Cycle. |
| **18. What is Error guessing and Error seeding?**  Error Guessing is a test case design technique where the tester has to guess what faults might occur and to design the tests to represent them.   Error Seeding is the process of adding known faults intentionally in a program for the reason of monitoring the rate of detection & removal and also to estimate the number of faults remaining in the program. |
| **19. Explain Compatibility testing with an example.**  Compatibility testing is to evaluate the application compatibility with the computing environment like Operating System, Database, Browser compatibility, backwards compatibility, computing capacity of the Hardware Platform and compatibility of the Peripherals. Example, If Compatibility testing is done on a Game application, before installing a game on a computer, its compatibility is checked with the computer specification that whether it is compatible with the computer having that much of specification or not. |
| **20. What is Test Harness?**  A test harness is a collection of software and test data required to test the application by running it in different testing condition like stress, load, data- driven, and monitoring its behavior and outputs. Test Harness contains two main parts:  - Test execution engine - Test script repository  Automation testing is the use of a tool to control the execution of tests and compare the actual results with the expected results. It also involves the setting up of test pre-conditions. |
| **21. Explain Statement coverage.**  Statement Coverage is a metric used in White Box Testing. Statement coverage is used to ensure that all the statement in the program code is executed at least once. The advantages of Statement Coverage are:  - Verifies that written code is correct. - Measures the quality of code written. - Determine the control flow of the program. - To Calculate Statement Coverage: - Statement Coverage = Statements Tested / Total No. of Statements. |
| **22. What are the types of testing?**  There are two types of testing:  - **Static testing:** Static testing is a technique used in the earlier phase of the development life cycle. The code error detection and execution of program is not concern in this type of testing. Also known as non-execution technique. The Verification of the product is performed in this testing technique like Code Reviews, Inspections, Walkthroughs are mostly done in this stage of testing.  - **Dynamic testing:** Dynamic Testing is concern with the execution of the software. This technique is used to test the dynamic behavior of the code. Most of the bugs are identified using this technique. These are the Validation activities. It uses different methodologies to perform testing like Unit Tests, Integration Tests, System Tests and Acceptance Testing, etc. |
| **23. Explain User acceptance testing.**  User Acceptance Testing (UAT) is performed by the end users on the applications before accepting the application.  **Alpha testing:** is performed by the IN-House developers. After alpha testing the software is handed for the Beta testing phase, for additional testing in an environment that is similar to the client environment.  **Beta testing:** is performed by the end user. So that they can make sure that the product is bug free or working as per the requirement. IN-house developers and software QA team perform alpha testing. The public, a few select prospective customers or the general public performs beta testing.  **Gamma Testing:** Gamma Testing is done when the software is ready for release with specified requirements. This testing is done directly by skipping all the in-house testing activities. |
| **24. What should be done after a bug is found?**  After finding the bug the first step is bug to be locked in bug report. Then this bug needs to be communicated and assigned to developers that can fix it. After the bug is fixes by the developer, fixes should be re-tested, and determinations made regarding requirements for regression testing to check that fixes didn't create problems elsewhere. |
| **25. What if the software is so buggy it can't really be tested at all?**  In this situation is for the testers to go through the process of reporting of bugs with the focus being on critical bugs. Since this type of problem can severely affect schedules, and indicates deeper problems in the software development process project managers should be notified, and provided with some documentation. |
| **26. What are the types of maintenance?**  **There are four types of maintenance. They are:**  - Corrective Maintenance - Adaptive Maintenance - Perfective Maintenance - Preventive Maintenance |
| **27. What are the advantages of waterfall model?**  **The advantages of the waterfall model are:**  - Simple to implement and required fewer amounts of resources. - After every phase output is generate. - Help in methods of analysis, design, coding, testing and maintenance. - Preferred in projects where quality is more important than schedule and cost. - Systematic and sequential model. - Proper documentation of the project. |
| **28. What is Rapid Application Development model (RAD)?**  The RAD model Rapid Application development (RAD) is incremental software development process models that focus on the development of the project in very short time. It is enhanced version of Waterfall model. It is proposed when requirements and solutions can be made independently system or software components, which is developed by different teams. After these smaller system components are developed, they are integrated to produce the large software system solution. |
| **29. What are the advantages of black box testing?**  **The advantages of this type of testing include:**  - Developer and tester are independent of each other. - The tester does not need knowledge of any programming languages. - The test is done from the point-of-view of the user. - Test cases can be designed when specifications are complete. - Testing helps to identify issues related to functional specifications. |
| **30. What is software review?**  A software review can be defined as a filter for the software engineering process. The purpose of any review is to discover errors in the analysis, design, and coding, testing and implementation phases of the software development cycle. The other purpose of a review is to see whether procedures are applied uniformly and in a manageable manner. It is used to check the process followed to develop the software is right. |
| **31. What is reverse engineering?**  By analyzing a final product the process of recreating a design is known as reverse engineering. Reverse engineering is the process followed in order to find difficult, unknown, and hidden information about a software system. It is important when software products lack proper documentation, and are highly unstructured, or their structure has degraded through a series of maintenance efforts. Maintenance activities cannot be performed without a complete understanding of the software system. |
| **32. What is data flow diagram?**  The Data Flow Diagram gives us information of the flow of data within the application.  - The DFD can be used to analyze the design of the application. - It is a graphical representation of the structure of the data. - A developer draws context level DFD first showing interaction between the different components of the application. - DFD help in developing the software by clarifying the requirements and major functionalities. - DFDs show the flow of data through a system. - It is an important modeling tool that allows us to picture a system as a network of functional processes. |
| **33. What is exploratory testing?**  **Exploratory testing:** means testing an application without a test plan and test script. In exploring testing test explore the application on the basis on his knowledge. The tester has no knowledge about the application previously. He explores the application like an end user and try to use it. While using the application his main motive is to find the bugs which are in the application. |
| **34. What is compatibility testing?**  Compatibility testing is a type of testing used to find out the compatibility between the application and platform on which application works, web browsers, hardware, operating systems etc. Good software must be compatible with different hardware, web browser and database. |
| **35. What is SRS and BRS document?**  **Software Requirements Specification (SRS)** is documented form of the requirement of the customer. It consists of all requirement of the customer regarding that software to be developed. The SRS document work as agreement between the company and the customer consisting of all functional and non functional requirements.  **Business Requirement Specification (BRS)** are the requirements as described by the business people. The business tells “what” they want for the application to do. In simple word BRS contain the functional requirement of the application. |
| **36. Can you explain V model in manual testing?**  **V model:** it is enhanced version of waterfall model where each level of the development lifecycle is verified before moving to next level. In this testing starts at the very beginning. By testing we mean verification by means of reviews and inspections, static testing. Each level of the development life - cycle has a corresponding test plan. A test plan is developed to prepare for the testing of the products of that phase. Be developing the test plans, we can also define the expected results for testing of the products for that level as well as defining the entry and exit criteria for each level. |
| **37. What is Concurrency Testing?**  **Concurrency Testing** is used to know the effects of using the software by different users at the same time. In this type of testing we have multiple users performing the exact same requests at the same time. It helps in identifying and measuring the problems in Response time, levels of locking and deadlocking in the application. For this we use Load runner to create VUGen (Virtual User Generator) is used to add the number of concurrent users and perform operation on the application on the same time. |
| **38. What is an inspection in software testing?**  An inspection is more formalized than a walk through. Inspection technique involves 3 to 8 team member consisting of a moderator, reader, and a recorder to take notes. The subject of the inspection is typically a document such as a requirements or a test plan, and the purpose is to find problems and see what is missing, most problems will be found during this preparation. The result of the inspection meeting should be a written report. It is one of the most cost effective methods of ensuring quality. |
| **39. A Form has four mandatory fields to be entered before you Submit. How many numbers of test cases are required to verify this? And what are they?**  **Five test cases are required to test:**  1. Enter the data in all the mandatory fields and submit, should not display error message. 2. Enter data in any two mandatory fields and summit, should issue an error message. 3. Do not enter in any of the fields should issue an error message. 4. If the fields accept only number, enter numbers in the fields and submit, should not issue an error message, try to enter only in two fields should issue an error message, and enter alphabets in two fields and number in other two fields it should issue an error message. 5. If the fields do not accept special characters, then enter the characters and submit it. |
| **40. What is Cyclomatic Complexity?**  **Cyclomatic complexity** is used to measure the complexity of the software using the control flow graph of the software. It is a graphical representation, consisting of following:  **NODE:** statement of the program is taken as node of the graph.  **Edges:** the flow of command is denoted by edges. Edges are used to connect two node , this show flow of control from one node to other node in the program.  Using this node and edges we calculate the complexity of the program. This determines the minimum number of inputs you need to test always to execute the program. |

**Q #1) What is Automation Testing?**

Automation testing or Test Automation is a process of automating the manual process to test the application/system under test. Automation testing involves use to a separate testing tool which lets you create test scripts which can be executed repeatedly and doesn’t require any manual intervention.

**Q #2) What are the benefits of Automation Testing?**

Benefits of Automation testing are:

1. Supports execution of repeated test cases
2. Aids in testing a large test matrix
3. Enables parallel execution
4. Encourages unattended execution
5. Improves accuracy thereby reducing human generated errors
6. Saves time and money

**Q #3) Why should Selenium be selected as a test tool?**

Selenium

1. is free and open source
2. have a large user base and helping communities
3. have cross Browser compatibility (Firefox, chrome, Internet Explorer, Safari etc.)
4. have great platform compatibility (Windows, Mac OS, Linux etc.)
5. supports multiple programming languages (Java, C#, Ruby, Python, Pearl etc.)
6. has fresh and regular repository developments
7. supports distributed testing

**Q #4) What is Selenium? What are the different Selenium components?**

Selenium is one of the most popular automated testing suites. Selenium is designed in a way to support and encourage automation testing of functional aspects of web based applications and a wide range of browsers and platforms. Due to its existence in the open source community, it has become one of the most accepted tools amongst the testing professionals.

Selenium is not just a single tool or a utility, rather a package of several testing tools and for the same reason it is referred to as a Suite. Each of these tools is designed to cater different testing and test environment requirements.

The suite package constitutes of the following sets of tools:

* [**Selenium Integrated Development Environment (IDE)**](http://www.softwaretestinghelp.com/selenium-ide-download-and-installation-selenium-tutorial-2/) – Selenium IDE is a record and playback tool. It is distributed as a Firefox Plugin.
* **Selenium Remote Control (RC)** – Selenium RC is a server that allows user to create test scripts in a desired programming language. It also allows executing test scripts within the large spectrum of browsers.
* [**Selenium WebDriver**](http://www.softwaretestinghelp.com/selenium-webdriver-selenium-tutorial-8/) – WebDriver is a different tool altogether that has various advantages over Selenium RC. WebDriver directly communicates with the web browser and uses its native compatibility to automate.
* [**Selenium Grid**](http://www.softwaretestinghelp.com/selenium-grid-selenium-tutorial-29/) – Selenium Grid is used to distribute your test execution on multiple platforms and environments concurrently.

**Q #5) What are the testing types that can be supported by Selenium?**

Selenium supports the following types of testing:

1. Functional Testing
2. Regression Testing

**Q #6) What are the limitations of Selenium?**

Following are the limitations of Selenium:

* Selenium supports testing of only web based applications
* Mobile applications cannot be tested using Selenium
* Captcha and Bar code readers cannot be tested using Selenium
* Reports can only be generated using third party tools like TestNG or Junit.
* As Selenium is a free tool, thus there is no ready vendor support though the user can find numerous helping communities.
* User is expected to possess prior programming language knowledge.

**Q #7)** **What is the difference between Selenium IDE, Selenium RC and WebDriver?**

| **Feature** | **Selenium IDE** | **Selenium RC** | **WebDriver** |
| --- | --- | --- | --- |
|  |  |  |  | |
| Browser Compatibility | Selenium IDE comes as a Firefox plugin, thus it supports only Firefox | Selenium RC supports a varied range of versions of Mozilla Firefox, Google Chrome, Internet Explorer and Opera | WebDriver supports a varied range of versions of Mozilla Firefox, Google Chrome, Internet Explorer and Opera. Also supports HtmlUnitDriver which is a GUI less or headless browser. | |
| Record and Playback | Selenium IDE supports record and playback feature | Selenium RC doesn't supports record and playback feature | WebDriver doesn't support record and playback feature | |
| Server Requirement | Selenium IDE doesn't require any server to be started before executing the test scripts | Selenium RC requires server to be started before executing the test scripts | WebDriver doesn't require any server to be started before executing the test scripts | |
| Architecture | Selenium IDE is a Javascript based framework | Selenium RC is a JavaScript based Framework | WebDriver uses the browser's native compatibility to automation | |
| Object Oriented | Selenium IDE is not an object oriented tool | Selenium RC is semi object oriented tool | WebDriver is a purely object oriented tool | |
| Dynamic Finders (for locating web elements on a webpage) | Selenium IDE doesn't support dynamic finders | Selenium RC doesn't support dynamic finders | WebDriver supports dynamic finders | |
| Handling Alerts, Navigations, Dropdowns | Selenium IDE doesn't explicitly provides aids to handle alerts, navigations, dropdowns | Selenium RC doesn't explicitly provides aids to handle alerts, navigations, dropdowns | WebDriver offers a wide range of utilities and classes that helps in handling alerts, navigations, and dropdowns efficiently and effectively. | |
| WAP (iPhone/Android) Testing | Selenium IDE doesn't support testing of iPhone/Andriod applications | Selenium RC doesn't support testing of iPhone/Andriod applications | WebDriver is designed in a way to efficiently support testing of iPhone/Android applications. The tool comes with a large range of drivers for WAP based testing. For example, AndroidDriver, iPhoneDriver | |
| Listener Support | Selenium IDE doesn't support listeners | Selenium RC doesn't support listeners | WebDriver supports the implementation of Listeners | |
| Speed | Selenium IDE is fast as it is plugged in with the web-browser that launches the test. Thus, the IDE and browser communicates directly | Selenium RC is slower than WebDriver as it doesn't communicates directly with the browser; rather it sends selenese commands over to Selenium Core which in turn communicates with the browser. | WebDriver communicates directly with the web browsers. Thus making it much faster. | |

**Q #8) When should I use Selenium IDE?**

Selenium IDE is the simplest and easiest of all the tools within the Selenium Package. Its record and playback feature makes it exceptionally easy to learn with minimal acquaintances to any programming language. Selenium IDE is an ideal tool for a naïve user.

**Q #9) What is Selenese?**

Selenese is the language which is used to write test scripts in Selenium IDE.

**Q #10)** **What are the different types of locators in Selenium?**

Locator can be termed as an address that identifies a web element uniquely within the webpage. Thus, to identify web elements accurately and precisely we have [different types of locators in Selenium](http://www.softwaretestinghelp.com/using-selenium-xpath-and-other-locators-selenium-tutorial-5/):

* ID
* ClassName
* Name
* TagName
* LinkText
* PartialLinkText
* Xpath
* CSS Selector
* DOM

**Q #11)** **What is difference between assert and verify commands?**

**Assert:**Assert command checks whether the given condition is true or false. Let’s say we assert whether the given element is present on the web page or not. If the condition is true then the program control will execute the next test step but if the condition is false, the execution would stop and no further test would be executed.

**Verify:**Verify command also checks whether the given condition is true or false. Irrespective of the condition being true or false, the program execution doesn’t halts i.e. any failure during verification would not stop the execution and all the test steps would be executed.

**Q #12) What is an Xpath?**

[Xpath](http://www.softwaretestinghelp.com/using-selenium-xpath-and-other-locators-selenium-tutorial-5/) is used to locate a web element based on its XML path. XML stands for Extensible Markup Language and is used to store, organize and transport arbitrary data. It stores data in a key-value pair which is very much similar to HTML tags. Both being markup languages and since they fall under the same umbrella, Xpath can be used to locate HTML elements.

The fundamental behind locating elements using Xpath is the traversing between various elements across the entire page and thus enabling a user to find an element with the reference of another element.

**Q #13) What is the difference between “/” and “//” in Xpath?**

**Single Slash “/” –**Single slash is used to create Xpath with absolute path i.e. the xpath would be created to start selection from the document node/start node.

**Double Slash “//” –** Double slash is used to create Xpath with relative path i.e. the xpath would be created to start selection from anywhere within the document.

**Q #14) What is Same origin policy and how it can be handled?**

The problem of same origin policy disallows to access the DOM of a document from an origin that is different from the origin we are trying to access the document.

Origin is a sequential combination of scheme, host and port of the URL. For example, for a URL http:// http://www.softwaretestinghelp.com/resources/, the origin is a combination of http, softwaretestinghelp.com, 80 correspondingly.

Thus the Selenium Core (JavaScript Program) cannot access the elements from an origin that is different from where it was launched. For Example, if I have launched the JavaScript Program from “http://www.softwaretestinghelp.com”, then I would be able to access the pages within the same domain such as “http://www.softwaretestinghelp.com/resources” or “http://www.softwaretestinghelp.com/istqb-free-updates/”. The other domains like google.com, seleniumhq.org would no more be accessible.

So, In order to handle same origin policy, Selenium Remote Control was introduced.

**Q #15)** **When should I use Selenium Grid?**

Selenium Grid can be used to execute same or different test scripts on multiple platforms and browsers concurrently so as to achieve distributed test execution, testing under different environments and saving execution time remarkably.

**Q #16) What do we mean by Selenium 1 and Selenium 2?**

Selenium RC and WebDriver, in a combination are popularly known as Selenium 2. Selenium RC alone is also referred as Selenium 1.

**Q #17) Which is the latest Selenium tool?**

WebDriver

**Q #18) How do I launch the browser using WebDriver?**

The following syntax can be used to launch Browser:  
*WebDriver driver =****new****FirefoxDriver();*  
*WebDriver driver =****new****ChromeDriver();*  
*WebDriver driver =****new****InternetExplorerDriver();*

**Q #19) What are the different types of Drivers available in WebDriver?**

The different drivers available in WebDriver are:

* FirefoxDriver
* InternetExplorerDriver
* ChromeDriver
* SafariDriver
* OperaDriver
* AndroidDriver
* IPhoneDriver
* HtmlUnitDriver

**Q #20) What are the different types of waits available in WebDriver?**

There are two [types of waits available in WebDriver](http://www.softwaretestinghelp.com/selenium-webdriver-waits-selenium-tutorial-15/):

1. Implicit Wait
2. Explicit Wait

**Implicit Wait:**Implicit waits are used to provide a default waiting time (say 30 seconds) between each consecutive test step/command across the entire test script. Thus, subsequent test step would only execute when the 30 seconds have elapsed after executing the previous test step/command.

**Explicit Wait:** Explicit waits are used to halt the execution till the time a particular condition is met or the maximum time has elapsed. Unlike Implicit waits, explicit waits are applied for a particular instance only.

**Q #21)** **How to type in a textbox using Selenium?**

User can use sendKeys(“String to be entered”) to enter the string in the textbox.

**Syntax:**  
*WebElement username = drv.findElement(By.id(“Email”));*  
*// entering username*  
*username.sendKeys(“sth”);*

**Q #22)** **How can you find if an element in displayed on the screen?**

WebDriver facilitates the user with the following methods to check the visibility of the web elements. These web elements can be buttons, drop boxes, checkboxes, radio buttons, labels etc.

1. isDisplayed()
2. isSelected()
3. isEnabled()

**Syntax:**

**isDisplayed():**  
***boolean****buttonPresence = driver.findElement(By.id(“gbqfba”)).isDisplayed();*

**isSelected():**  
***boolean****buttonSelected = driver.findElement(By.id(“gbqfba”)).isDisplayed();*

**isEnabled():**  
***boolean****searchIconEnabled = driver.findElement(By.id(“gbqfb”)).isEnabled();*

**Q #23)** **How can we get a text of a web element?**

Get command is used to retrieve the inner text of the specified web element. The command doesn’t require any parameter but returns a string value. It is also one of the extensively used commands for verification of messages, labels, errors etc displayed on the web pages.

**Syntax:**  
*String Text = driver.findElement(By.id(“Text”)).getText();*

**Q #24) How to select value in a dropdown?**

Value in the drop down can be selected using WebDriver’s Select class.

**Syntax:**

**selectByValue:**  
*Select selectByValue =****new****Select(driver.findElement(By.id(“SelectID\_One”)));*  
*selectByValue.selectByValue(“greenvalue”);*

**selectByVisibleText:**  
*Select selectByVisibleText =****new****Select (driver.findElement(By.id(“SelectID\_Two”)));*  
*selectByVisibleText.selectByVisibleText(“Lime”);*

**selectByIndex:**  
*Select selectByIndex =****new****Select(driver.findElement(By.id(“SelectID\_Three”)));*  
*selectByIndex.selectByIndex(2);*

**Q #25) What are the different types of navigation commands?**

Following are the [navigation commands](http://www.softwaretestinghelp.com/selenium-webdriver-waits-selenium-tutorial-15/):  
**navigate().back()** – The above command requires no parameters and takes back the user to the previous webpage in the web browser’s history.

**Sample code:**  
*driver.navigate().back();*

**navigate().forward()** – This command lets the user to navigate to the next web page with reference to the browser’s history.

**Sample code:**  
*driver.navigate().forward();*

**navigate().refresh()** – This command lets the user to refresh the current web page there by reloading all the web elements.

**Sample code:**  
*driver.navigate().refresh();*

**navigate().to()** – This command lets the user to launch a new web browser window and navigate to the specified URL.

**Sample code:**  
*driver.navigate().to(“https://google.com”);*

**Q #26) How to click on a hyper link using linkText?**

*driver.findElement(By.linkText(“Google”)).click();*

The command finds the element using link text and then click on that element and thus the user would be re-directed to the corresponding page.

The above mentioned link can also be accessed by using the following command.

*driver.findElement(By.partialLinkText(“Goo”)).click();*

The above command find the element based on the substring of the link provided in the parenthesis and thus partialLinkText() finds the web element with the specified substring and then clicks on it.

**Q #27)** **How to**[**handle frame in WebDriver**](http://www.softwaretestinghelp.com/selenium-tutorial-18/)**?**

An inline frame acronym as iframe is used to insert another document with in the current HTML document or simply a web page into a web page by enabling nesting.

**Select iframe by id**  
*driver.switchTo().frame(“ID of the frame“);*

**Locating iframe using tagName**  
*driver.switchTo().frame(driver.findElements(By.tagName(“iframe”).get(0));*

**Locating iframe using index**

**frame(index)**  
*driver.switchTo().frame(0);*

**frame(Name of Frame)**  
*driver.switchTo().frame(“name of the frame”);*

**frame(WebElement element)**  
**Select Parent Window**  
*driver.switchTo().defaultContent();*

**Q #28) When do we use findElement() and findElements()?**

**findElement():**findElement() is used to find the first element in the current web page matching to the specified locator value. Take a note that only first matching element would be fetched.

**Syntax:**

*WebElement element =driver.findElements(By.xpath(“//div[@id=’example’]//ul//li”));*  
**findElements():**findElements() is used to find all the elements in the current web page matching to the specified locator value. Take a note that all the matching elements would be fetched and stored in the list of WebElements.

**Syntax:**  
*List <WebElement> elementList =driver.findElements(By.xpath(“//div[@id=’example’]//ul//li”));*

**Q #29)** **How to find more than one web element in the list?**

At times, we may come across elements of same type like multiple hyperlinks, images etc arranged in an ordered or unordered list. Thus, it makes absolute sense to deal with such elements by a single piece of code and this can be done using WebElement List.

**Sample Code**

|  |  |
| --- | --- |
| 1 | // Storing the list |
| 2 | List <WebElement> elementList = driver.findElements(By.xpath("//div[@id='example']//ul//li")); | |

|  |  |
| --- | --- |
| 3 | // Fetching the size of the list |
| 4 | int listSize = elementList.size(); | |

|  |  |  |
| --- | --- | --- |
| 5 | for (int i=0; i<listSize; i++) | |
| 6 | { |

|  |  |  |
| --- | --- | --- |
| 7 | // Clicking on each service provider link | |
| 8 | serviceProviderLinks.get(i).click(); |

|  |  |  |  |
| --- | --- | --- | --- |
| 9 | // Navigating back to the previous page that stores link to service providers | | |
| 10 | | driver.navigate().back(); |

|  |  |
| --- | --- |
| 11 | } |

**Q #30) What is the difference between driver.close() and driver.quit command?**

**close()**: WebDriver’s close() method closes the web browser window that the user is currently working on or we can also say the window that is being currently accessed by the WebDriver. The command neither requires any parameter nor does is return any value.

**quit()**: Unlike close() method, quit() method closes down all the windows that the program has opened. Same as close() method, the command neither requires any parameter nor does is return any value.

**Q #31) Can Selenium handle windows based pop up?**

Selenium is an automation testing tool which supports only web application testing. Therefore, windows pop up cannot be handled using Selenium.

**Q #32) How can we handle web based pop up?**

WebDriver offers the users with a very efficient way to [handle these pop ups using Alert interface](http://www.softwaretestinghelp.com/handle-alerts-popups-selenium-webdriver-selenium-tutorial-16/). There are the four methods that we would be using along with the Alert interface.

* void dismiss() – The accept() method clicks on the “Cancel” button as soon as the pop up window appears.
* void accept() – The accept() method clicks on the “Ok” button as soon as the pop up window appears.
* String getText() – The getText() method returns the text displayed on the alert box.
* void sendKeys(String stringToSend) – The sendKeys() method enters the specified string pattern into the alert box.

**Syntax:**  
*// accepting javascript alert*  
*Alert alert = driver.switchTo().alert();*  
*alert.accept();*

**Q #33) How can we handle windows based pop up?**

Selenium is an automation testing tool which supports only web application testing, that means, it doesn’t support testing of windows based applications. However Selenium alone can’t help the situation but along with some third party intervention, this problem can be overcome. There are several third party tools available for handling window based pop ups along with the selenium like AutoIT, Robot class etc.

**Q #34) How to assert title of the web page?**

*//verify the title of the web page*  
*assertTrue(“The title of the window is incorrect.”,driver.getTitle().equals(“Title of the page”));*

**Q #35) How to mouse hover on a web element using WebDriver?**

WebDriver offers a wide range of interaction utilities that the user can exploit to automate mouse and keyboard events. Action Interface is one such utility which simulates the single user interactions.

Thus, In the following scenario, we have used Action Interface to mouse hover on a drop down which then opens a list of options.

**Sample Code:**

|  |  |
| --- | --- |
| 1 | // Instantiating Action Interface |
| 2 | Actions actions=new Actions(driver); | |

|  |  |
| --- | --- |
| 3 | // howering on the dropdown |
| 4 | actions.moveToElement(driver.findElement(By.id("id of the dropdown"))).perform(); | |

|  |  |
| --- | --- |
| 5 | // Clicking on one of the items in the list options |
| 6 | WebElement subLinkOption=driver.findElement(By.id("id of the sub link")); | |

|  |  |
| --- | --- |
| 7 | subLinkOption.click(); |

**Q #36) How to retrieve css properties of an element?**

The values of the css properties can be retrieved using a get() method:

**Syntax:**  
*driver.findElement(By.id(“id“)).getCssValue(“name of css attribute”);*  
*driver.findElement(By.id(“id“)).getCssValue(“font-size”);*

**Q #37) How to capture screenshot in WebDriver?**

|  |  |
| --- | --- |
| 1 | import org.junit.After; |
| 2 | import org.junit.Before; | |

|  |  |  |
| --- | --- | --- |
| 3 | import org.junit.Test; | |
| 4 | import java.io.File; |

|  |  |
| --- | --- |
| 5 | import java.io.IOException; |
| 6 | import org.apache.commons.io.FileUtils; | |

|  |  |
| --- | --- |
| 7 | import org.openqa.selenium.OutputType; |
| 8 | import org.openqa.selenium.TakesScreenshot; | |

|  |  |  |
| --- | --- | --- |
| 9 | import org.openqa.selenium.WebDriver; | |
| 10 | | import org.openqa.selenium.firefox.FirefoxDriver; | |

|  |  |
| --- | --- |
| 11 |  |
| 12 | public class CaptureScreenshot { | |

|  |  |  |
| --- | --- | --- |
| 13 | WebDriver driver; | |
| 14 | @Before |

|  |  |  |
| --- | --- | --- |
| 15 | public void setUp() throws Exception { | |
| 16 | driver = new FirefoxDriver(); |

|  |  |  |
| --- | --- | --- |
| 17 | driver.get("https://google.com"); | |
| 18 | } |

|  |  |
| --- | --- |
| 19 | @After |
| 20 | public void tearDown() throws Exception { | |

|  |  |  |
| --- | --- | --- |
| 21 | driver.quit(); | |
| 22 | } |

|  |  |
| --- | --- |
| 23 |  |
| 24 | @Test | |

|  |  |
| --- | --- |
| 25 | public void test() throws IOException { |
| 26 | // Code to capture the screenshot | |

|  |  |  |
| --- | --- | --- |
| 27 | File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE); | |
| 28 | // Code to copy the screenshot in the desired location |

|  |  |  |  |
| --- | --- | --- | --- |
| 29 | FileUtils.copyFile(scrFile, newFile("C:\\CaptureScreenshot\\google.jpg")); | | |
| 30 | | } |

|  |  |
| --- | --- |
| 31 | } |

**Q #38) What is Junit?**

[Junit](http://www.softwaretestinghelp.com/selenium-junit-framework-selenium-tutorial-11/) is a unit testing framework introduced by Apache. Junit is based on Java.

**Q #39) What are Junit annotations?**

Following are the Junit Annotations:

* **@Test:**Annotation lets the system know that the method annotated as @Test is a test method. There can be multiple test methods in a single test script.
* **@Before:**Method annotated as @Before lets the system know that this method shall be executed every time before each of the test method.
* **@After:**Method annotated as @After lets the system know that this method shall be executed every time after each of the test method.
* **@BeforeClass:**Method annotated as @BeforeClass lets the system know that this method shall be executed once before any of the test method.
* **@AfterClass:**Method annotated as @AfterClass lets the system know that this method shall be executed once after any of the test method.
* **@Ignore:**Method annotated as @Ignore lets the system know that this method shall not be executed.

**Q #40)** **What is TestNG and how is it better than Junit?**

[TestNG](http://www.softwaretestinghelp.com/testng-framework-selenium-tutorial-12/) is an advance framework designed in a way to leverage the benefits by both the developers and testers. With the commencement of the frameworks, JUnit gained an enormous popularity across the Java applications, Java developers and Java testers with remarkably increasing the code quality. Despite being easy to use and straightforward, JUnit has its own limitations which give rise to the need of bringing TestNG into the picture. TestNG is an open source framework which is distributed under the Apache software License and is readily available for download.

TestNG with WebDriver provides an efficient and effective test result format that can in turn be shared with the stake holders to have a glimpse on the product’s/application’s health thereby eliminating the drawback of WebDriver’s incapability to generate test reports. TestNG has an inbuilt exception handling mechanism which lets the program to run without terminating unexpectedly.

There are various advantages that make TestNG superior to JUnit. Some of them are:

* Added advance and easy annotations
* Execution patterns can set
* Concurrent execution of test scripts
* Test case dependencies can be set

**Q #41)** **How to set test case priority in TestNG?**

**Setting Priority in TestNG**

**Code Snippet**

|  |  |
| --- | --- |
| 1 | package TestNG; |
| 2 | import org.testng.annotations.\*; | |

|  |  |  |
| --- | --- | --- |
| 3 | public class SettingPriority { | |
| 4 | @Test(priority=0) |

|  |  |  |
| --- | --- | --- |
| 5 | public void method1() { | |
| 6 | } |

|  |  |
| --- | --- |
| 7 | @Test(priority=1) |
| 8 | public void method2() { | |

|  |  |  |
| --- | --- | --- |
| 9 | } | |
| 10 | | @Test(priority=2) | |

|  |  |  |
| --- | --- | --- |
| 11 | public void method3() { | |
| 12 | } |

|  |  |
| --- | --- |
| 13 | } |

**Test Execution Sequence:**

1. Method1
2. Method2
3. Method3

**Q #42) What is a framework?**

Framework is a constructive blend of various guidelines, coding standards, concepts, processes, practices, project hierarchies, modularity, reporting mechanism, test data injections etc. to pillar automation testing.

**Q #43)** **What are the advantages of Automation framework?**

**Advantage of**[**Test Automation framework**](http://www.softwaretestinghelp.com/test-automation-frameworks-selenium-tutorial-20/)

* Reusability of code
* Maximum coverage
* Recovery scenario
* Low cost maintenance
* Minimal manual intervention
* Easy Reporting

**Q #44) What are the different types of frameworks?**

**Below are the different types of frameworks:**

1. **Module Based Testing Framework:** The framework divides the entire “Application Under Test” into number of logical and isolated modules. For each module, we create a separate and independent test script. Thus, when these test scripts taken together builds a larger test script representing more than one module.
2. **Library Architecture Testing Framework:** The basic fundamental behind the framework is to determine the common steps and group them into functions under a library and call those functions in the test scripts whenever required.
3. Data Driven Testing Framework: Data Driven Testing Framework helps the user segregate the test script logic and the test data from each other. It lets the user store the test data into an external database. The data is conventionally stored in “Key-Value” pairs. Thus, the key can be used to access and populate the data within the test scripts.
4. **Keyword Driven Testing Framework:** The Keyword driven testing framework is an extension to Data driven Testing Framework in a sense that it not only segregates the test data from the scripts, it also keeps the certain set of code belonging to the test script into an external data file.
5. **Hybrid Testing Framework:** Hybrid Testing Framework is a combination of more than one above mentioned frameworks. The best thing about such a setup is that it leverages the benefits of all kinds of associated frameworks.
6. **Behavior Driven Development Framework:** Behavior Driven Development framework allows automation of functional validations in easily readable and understandable format to Business Analysts, Developers, Testers, etc.

**Q #45) How can I read test data from excels?**

Test data can efficiently be read from excel using JXL or POI API. [See detailed tutorial here](http://www.softwaretestinghelp.com/selenium-framework-design-selenium-tutorial-21/).

**Q #46) What is the difference between POI and jxl jar?**

| **#** | **JXL jar** | **POI jar** |
| --- | --- | --- |
| 1 | JXL supports “.xls” format i.e. binary based format. JXL doesn’t support Excel 2007 and “.xlsx” format i.e. XML based format | POI jar supports all of these formats |
| 2 | JXL API was last updated in the year 2009 | POI is regularly updated and released |
| 3 | The JXL documentation is not as comprehensive as that of POI | POI has a well prepared and highly comprehensive documentation |
| 4 | JXL API doesn’t support rich text formatting | POI API supports rich text formatting |
| 5 | JXL API is faster than POI API | POI API is slower than JXL API |

**Q #47)** **What is the difference between Selenium and QTP?**

| **Feature** | **Selenium** | **Quick Test Professional (QTP)** |
| --- | --- | --- |
| Browser Compatibility | Selenium supports almost all the popular browsers like Firefox, Chrome, Safari, Internet Explorer, Opera etc | QTP supports Internet Explorer, Firefox and Chrome. QTP only supports Windows Operating System |
| Distribution | Selenium is distributed as an open source tool and is freely available | QTP is distributed as a licensed tool and is commercialized |
| Application under Test | Selenium supports testing of only web based applications | QTP supports testing of both the web based application and windows based application |
| Object Repository | Object Repository needs to be created as a separate entity | QTP automatically creates and maintains Object Repository |
| Language Support | Selenium supports multiple programming languages like Java, C#, Ruby, Python, Perl etc | QTP supports only VB Script |
| Vendor Support | As Selenium is a free tool, user would not get the vendor’s support in troubleshooting issues | Users can easily get the vendor’s support in case of any issue |

**Q #48) Can WebDriver test Mobile applications?**

WebDriver cannot test Mobile applications. WebDriver is a web based testing tool, therefore applications on the mobile browsers can be tested.

**Q #49) Can captcha be automated?**

No, captcha and bar code reader cannot be automated.

**Q #50) What is Object Repository? How can we create Object Repository in Selenium?**

Object Repository is a term used to refer to the collection of web elements belonging to Application Under Test (AUT) along with their locator values. Thus, whenever the element is required within the script, the locator value can be populated from the Object Repository. Object Repository is used to store locators in a centralized location instead of hard coding them within the scripts.

In Selenium, objects can be stored in an excel sheet which can be populated inside the script whenever required.

1. **Tell me some TestNG Annotations.**

**@Test,@Parameters,@Listeners,@BeforeSuite,@AfterSuite,@BeforeTest,@AfterTest,  
@DataProvider,@BeforeGroups,@AfterGroups,@BeforeClass,@AfterClass,  
@BeforeMethod,@AfterMethod,@Factory**

1. **What are desired capabilities?**

**Desired Capabilities help to set properties for the Web Driver. A typical use case would be to   
set the path for the Firefox Driver if your local installation doesn't correspond to the default   
settings.**

1. **Difference between Selenium RC and Selenium  
   Web driver.**

|  |  |
| --- | --- |
| **Selenium RC** | **Selenium Web driver** |
| **Selenium RC’s architecture is way more complicated.** | **Web Driver’s architecture is simpler than Selenium RC’s.** |
| **Selenium RC is slower since it uses a JavaScript program called Selenium Core. This Selenium Core is the one that directly controls the browser, not you.** | **Web Driver is faster than Selenium RC since it speaks directly to the browser uses the browser’s own engine to control it.** |
| **Selenium Core, just like other JavaScript codes, can access disabled elements.** | **Web Driver interacts with page elements in a more realistic way.** |
| **Selenium RC’s API is more matured but contains redundancies and often confusing commands.** | **Web Driver’s API is simpler than Selenium RC’s. It does not contain redundant and confusing commands.** |
| **Selenium RC cannot support the headless HtmlUnit browser. It needs a real, visible browser to operate on.** | **Web Driver can support the headless HtmlUnit browser.** |
| **Selenium RC Has Built-In Test Result Generator. Selenium RC automatically generates an HTML file of test results.** | **Web Driver has no built-in command that automatically generates a Test Results File.** |
| **Selenium RC needs the help of the RC Server in order to do so.** | **web Driver directly talks to the browser** |
| **Selenium RC can support new browsers** | **It cannot readily support new browsers** |

1. **Difference between Web driver listener and  
   TestNG Listener.**

**TestNG and Web driver Listener have different interfaces to implement and call them. They both   
modify respective behaviour. You can use Listeners in Annotation. Below 2 URL gives the  
detailed list of listener and their interfaces.**

1. **Describe your framework.**

1. **Which is the best way to locate an element?**

**Finding elements by ID is usually going to be the fastest option, because at its root, it eventually calls down to document.getElementById(), which is optimized by many browsers.**

**Finding elements by XPath is useful for finding elements using very complex selectors, and is the most flexible selection strategy, but it has the potential to be very slow, particularly in IE. In IE 6, 7, or 8, finding by XPath can be an order of magnitude slower than doing the same in Firefox. IE provides no native XPath-over-HTML solution, so the project must use a JavaScript XPath implementation, and the JavaScript engine in legacy versions of IE really is that much slower.**

**If you have a need to find an element using a complex selector, I usually recommend using CSS Selectors, if possible. It's not quite as flexible as XPath, but will cover many of the same cases, without exhibiting the extreme performance penalty on IE that XPath can.**

1. **Why we refer Firefox driver to the web driver  
   inheritance.**

**web Driver driver = new FireFoxDriver();**

**WebDriver is an interface which contain several abstract methods such as get(...), findElamentBy(...) etc.**

**We simply create reference of web Driver and we can assign objects (Firefox driver, CromeDriver, IEDriver, Andriod driver etc) to it.**

**Ex :**

**WebDriver driver = new FireFoxDriver();-----------(1)**

**If we are using (1) we can do the same thing by using**

**FireFoxDriver driver = new FireFoxDriver();---------(2)**

**We can use (1) and (2) for same purpose but if we want to switch to another browser in same program**

**then again we have to create the object of other class as for example**

**CromeDriver driver = new CromeDriver();.**

**creating object of several class is not good. So we create the reference of WebDriver and**

**we assign the objects of another class as for example**

**WebDriver driver; // it is created only one time in the program**

**driver = new FireFoxDriver();// any where in the program**

**driver = new CromeDriver(); // any where in the program**

1. **What are the features of TestNG?**

**TestNG is a testing framework designed to simplify a broad range of testing needs, from   
unit testing (testing a class in isolation of the others) to integration testing (testing entire   
systems made of several classes, several packages and even several external frameworks,  
such as application servers). You can use test suite,annotations, automatically generation  
of report and much more.**

1. **What is the difference between thread.Sleep()  
   and selenium. Set Speed ("2000")?**

**If the application is taking time to load the page then we use selenium.waitforpageload(" "). This command is doesn’t wait upto the given time whenever the page load is completed.**

**If the application is taking time to refresh the page, then we use Thread. Sleep ( ).it is a standard wait it simply wait to the given time.**

**selenium.setSpeed**

1. **Takes a single argument in string format**

**Ex: selenium.setSpeed("2000") - will wait for 2 seconds**

1. **Runs each command in after setSpeed delay by the number of milliseconds mentioned in set Speed.**

**thread.sleep**

1. **Takes a single argument in integer format**

**ex: thread. Sleep(2000) - will wait for 2 seconds**

1. **Waits for only once at the command given at sleep.**

1. **In what situation selenium finding element   
   get fails?**

* **· Element loading issue**
* **· Dynamic id of web element**

1. **What is the difference between "GET" and**

**"NAVIGATE" to open a web page in selenium   
web driver?**

**Get method will get a page to load or get page source or get text that's all whereas navigate  
will guide through the history like refresh, back, forward.For example if we want to move   
forward and do some functionality and back to the home page this can be achieved   
through navigate() only. driver.get will wait till the whole page gets loaded and driver.navigate   
will just redirect to that page and will not wait**

1. **Please tell me the difference b/w implicitly   
   Wait and Explicit wait.**

**Implicit Wait sets internally a timeout that will be used for all consecutive Web Element searches. It will try lookup the element again and again for the specified amount of time before throwing a NoSuchElementException if the element could not have been found. It does only this and can't be forced into anything else - it waits for elements to show up.**

**Explicit Wait or just Wait is a one-timer used by you for a particular search. It is more extendible in the means that you can set it up to wait for any condition you might like. Usually, you can use some of the prebuilt Expected Conditions to wait for elements to become clickable, visible, invisible, etc., or just write your own condition that suits your needs.**

1. **How we can retrieve the dynamically changing   
   Ids? When we login Facebook the login label's id   
   changes dynamically thus resulting in failure.**

**We have a hierarchy of locators and Facebook Is dynamic in nature,so we are not able to   
use "id" for identification for after that we have remaining 7 locator's for that :2. xpath ()..  
3. name..4. css.. 5. link text.. 6. partiallinktext...7.tag name. so u can use any one for  
identifying it. Most probably u can use "xpath" or "css-locator" and if there r tag then   
link text or partial-link text. it depend on u . But we never use id's in Ajax application  
because it’s not possible.**

**15.What is the difference between driver.Close()  
and driver.Quit () method?**

**Close() - It is used to close the browser or page currently which is having the focus.**

**Quit() - It is used to shut down the web driver instance or destroy the web driver instance  
(Close all the windows)**

1. **How to scroll web element?--not browser—**

**FirefoxProfile profile=new FirefoxProfile();**

**profile.setEnableNativeEvents(true);**

**WebDriver driver=new FirefoxDriver(profile);**

**driver.navigate("http://jqueryui.com/draggable/");**

**Thread.sleep(6000L);**

**WebElement element=driver.findElement(By.xpath("//div[@id='draggable']"));**

**Actions actn=new Actions(driver);**

**actn.dragAndDropBy(element, 50, 50).build().perform();**

**}**

1. **What is the basic use of Firefox profiles and   
   how can we use them using selenium?**

**A profile in Firefox is a collection of bookmarks, browser settings, extensions, passwords,   
and history; in short, all of your personal settings.**

**We use them to change user agent, changing default download directory, changing versions etc.**

1. **Customize the name of file going to be   
   downloaded?**

**You have to download AUTO IT.exe file and has to be install**

**and later you have create .au3 file (in this file you have to specify the commands in  
VB script like your file name, where have to save, it will be easy may be 3 or 4 steps )   
using AUTOIT...then right click the .au3 file you have to compile ....after that you will   
get the .exe file with the name of .au3 file ..In eclipse you will give the code like this**

**<----ProcessBuildder ps = new ProcessBuilder("path of the .exe file of au3") .start();--->**

1. **How to handle internationalisation through  
   web driver?**

**FirefoxProfile profile = new FirefoxProfile();**

**profile.set Preference("intl.accept\_languages","jp");**

**Web driver driver = new FirefoxDriver(profile); driver.get(google.com) will open google in   
Japanese Lang**

1. **How to overcome same origin policy through   
   web driver?**

* **· Proxy server.**

**DesiredCapabilities capability=new DesiredCapabilities.firefox();**

**capability.setCapability(CapabilityType.PROXY,"your desire proxy")**

**WebDriver driver=new FirefoxDriver(capability);**

1. **How to put text in Facebook search box using   
   selenium web driver.**

* **· driver.findElement(By.xpath("//div[contains(@class, '\_586i')]")).sendKeys("abc");**

1. **Difference between flex and flash application.**

**In flash there is no code just based on creativity(design) we will complete the   
work(time consuming process) whereas flex contain some small functions   
which is integrated with mxml,PHP..(no tool is there to develop**

**flex we want to use the properties of css and style sheet)**

1. **What is Error Collector in TestNG?   
   What is its use?**

**This class allows the collection of errors during the process of retrieving the   
test data for the test method parameters**

1. **How can we get the font size, font color,   
   font type used for a particular text on a web  
   page using Selenium web driver?**

**driver.findelement(By.Xpath("Xpath ").getcssvalue("font-size);**

**driver.findelement(By.Xpath("Xpath ").getcssvalue("font-colour);**

**driver.findelement(By.Xpath("Xpath ").getcssvalue("font-type);**

**driver.findelement(By.Xpath("Xpath ").getcssvalue("background-colour);**

1. **How to run tests in multiple browser parallel?  
   Is there any other option other than selenium grid?**

**You create a class with a method something like this:  
  
  
public class LaunchBrowser {  
  
WebDriver driver=null;  
  
  
// Pass parameter browser from test.xml  
@Parameters(“browser”)   
public void initiateBrowser(String browser){  
  
// compare browser to fire fox and then open firefox driver  
if(browser.equals(“Firefox”))  
{  
  
driver = new FirefoxDriver();  
}  
else  
{  
\ set path to the IE driver correctly here  
System.setProperty("webdriver.ie.driver", "\iexploredriver.exe");  
driver =new InternetExplorerDriver();  
}  
}  
  
Now create YourClassName class and call extend the above class something like this  
  
@Test  
public class YourClassName extends LaunchBrowser{  
  
public void gotoGoogle(){  
  
  
driver.get(“http://www.google.com");  
}  
}**

1. **How to prepare Customized html Report using  
   TestNG in hybrid framework.**

**Below are the 3 ways:**

* **Junit: with the help of ANT.**
* **TestNG: using inbuilt default.html to get the HTML report. Also XST reports from ANT,**

**Selenium, TestNG combination.**

* **Using our own customized reports using XSL jar for converting XML content to HTML.**

1. **“What’s the hierarchy of TestNG annotations?**

**Explain me about annotation hierarchy & execution  
order?**

1. **org.testng.annotations.Parameters (implements java.lang.annotation.Annotation)**
2. **org.testng.annotations.Listeners (implements java.lang.annotation.Annotation)**
3. **org.testng.annotations.Test (implements java.lang.annotation.Annotation)**
4. **org.testng.annotations.AfterMethod (implements java.lang.annotation.Annotation)**
5. **org.testng.annotations.BeforeTest (implements java.lang.annotation.Annotation)**
6. **org.testng.annotations.BeforeMethod (implements java.lang.annotation.Annotation)**
7. **org.testng.annotations.Optional (implements java.lang.annotation.Annotation)**
8. **org.testng.annotations.AfterTest (implements java.lang.annotation.Annotation)**
9. **org.testng.annotations.Guice (implements java.lang.annotation.Annotation)**
10. **org.testng.annotations.BeforeGroups (implements java.lang.annotation.Annotation)**
11. **org.testng.annotations.ExpectedExceptions (implements java.lang.annotation.Annotation)**
12. **org.testng.annotations.TestInstance (implements java.lang.annotation.Annotation)**
13. **org.testng.annotations.NoInjection (implements java.lang.annotation.Annotation)**
14. **org.testng.annotations.AfterSuite (implements java.lang.annotation.Annotation)**
15. **org.testng.annotations.AfterClass (implements java.lang.annotation.Annotation)**
16. **org.testng.annotations.AfterGroups (implements java.lang.annotation.Annotation)**
17. **org.testng.annotations.DataProvider (implements java.lang.annotation.Annotation)**
18. **org.testng.annotations.BeforeSuite (implements java.lang.annotation.Annotation)**
19. **org.testng.annotations.BeforeClass (implements java.lang.annotation.Annotation)**
20. **org.testng.annotations.Factory (implements java.lang.annotation.Annotation)**
21. **org.testng.annotations.Configuration (implements java.lang.annotation.Annotation)**
22. **org.testng.annotations.ObjectFactory (implements java.lang.annotation.Annotation)**

1. **How the TestNG interacts with Selenium Core?**

**Explain me steps and internal architecture?"**

|  |  |
| --- | --- |
|  |  |

**What is TestNG?**

**So far we had been doing Selenium tests without generating a proper format for the test results. From this point on, we shall tackle how to make these reports using a test**[**framework**](http://www.guru99.com/creating-automation-frameworks-with-qtp.html)**called TestNG.**

**TestNG is a**[**testing**](http://www.guru99.com/software-testing.html)**framework that overcomes the limitations of another popular testing framework called JUnit. The "NG" means "Next Generation". Most Selenium users use this more than JUnit because of its advantages. There are so many features of TestNG, but we will only focus on the most important ones that we can use in Selenium.Advantages of TestNG over JUnit**

**There are three major advantages of TestNG over JUnit:**

* **Annotations are easier to understand**
* **Test cases can be grouped more easily**
* **Parallel testing is possible**

1. **Is it possible test web services using selenium?**

**Using Jmeter we can test how one website is talking to each other means time taken to   
send data, feeds, messages from one website to other website. Jmeter does a nice job   
of doubling for performance and api tests.**

1. **How to refresh a page without using context   
   click?**

**1.Using sendKeys.Keys method**

**2.Using navigate.refresh() method**

**3.Using navigate.refresh() method**

**4.Using get() method**

**5.Using sendKeys() method**

**1.Using sendKeys.Keys method**

**driver.get("https://accounts.google.com/SignUp");**

**driver.findElement(By.id("firstname-placeholder")).sendKeys(Keys.F5);**

**2.Using navigate.refresh() method**

**driver.get("http://ruchi-myseleniumblog.blogspot.in/2013/12/100-selenium-interview-questions.html");**

**driver.navigate().refresh();**

**3.Using navigate.to() method**

**driver.get("http://ruchi-myseleniumblog.blogspot.in/2014/01/selenium-hybrid-framework-using.html");**

**driver.navigate().to(driver.getCurrentUrl());**

**4.Using get() method**

**driver.get("http://ruchi-myseleniumblog.blogspot.in/2013/12/basic-core-java-interview-questions.html");**

**driver.get(driver.getCurrentUrl());**

**5.Using sendKeys() method**

**driver.get("https://accounts.google.com/SignUp");**

**driver.findElement(By.id("firstname-placeholder")).sendKeys("\uE035");**

1. **Can u send a code for printing in selenium?**

**There are two cases:**

**Case1. Any hyperlink/button on a web page, n clicking that link/button a print dialog box   
opens. (Performing an action on web page)**

**Case2.or do u want to open print dialog box within ur own script, not by performing any   
action on web page.**

**So If Case 1: just a call for WebElement.click() event will work to open it.**

**If Case 2: Call a Printer Job object (Use Awt API).**

**For code: Google it.**

1. **How to find broken images in a page using   
   Selenium Web driver.**
2. **Get xpath and then using tag name; get all the links in the page**
3. **Click on each and every link in the page**
4. **In the target page title, look for 404/500 error.**

**How to find broken images in a page using Selenium**

**package programs;  
  
import java.util.List;  
import org.apache.http.HttpResponse;  
import org.apache.http.client.methods.HttpGet;  
import org.apache.http.impl.client.DefaultHttpClient;  
import org.openqa.selenium.By;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.WebElement;  
import org.openqa.selenium.firefox.FirefoxDriver;  
  
public class findbrokenimages {  
static int invalidimg;  
static WebDriver driver ;  
public static void main(String[] args) {  
try {  
driver = new FirefoxDriver();  
driver.get("http://ruchi-myseleniumblog.blogspot.in");  
invalidimg = 0;  
List allImages = driver.findElements(By.tagName("img"));  
System.out.println("Total images are " + allImages.size());  
for (int i = 0; i < allImages.size(); i++) {  
WebElement img = (WebElement) allImages.get(i);  
if (img != null) {  
verifyimgActive(img);  
}  
}  
  
System.out.println("Total invalid images are " + invalidimg);  
driver.quit();  
} catch (Exception e) {  
e.printStackTrace();  
System.out.println(e.getMessage());  
}  
}  
  
  
  
public static void verifyimgActive(WebElement img) {  
try {  
HttpResponse response = new DefaultHttpClient().execute(new HttpGet(img.getAttribute("src")));  
if (response.getStatusLine().getStatusCode() != 200)  
invalidimg++;  
}  
catch (Exception e) {  
e.printStackTrace();  
}  
}  
}**

1. **How to handle Ajax popup window?**

**By using getWindowHandles() and obj.switchTo.window(windowid) we can handle popups using   
explicit wait and driver.swtchT0.window("name") commands for your requirements.**

1. **How to handle auto complete box in web driver?**

**How to handle autocomplete box in web driver**

**How to handle autocomplete box in web driver?**

**driver.findElement(By.id("your searchBox")).sendKeys("your partial keyword");**

**Thread.sleep(3000);**

**List <WebElement> listItems = driver.findElements(By.xpath("your list item locator"));**

**listItems.get(0).click();**

**driver.findElement(By.id("your searchButton")).click();**

1. **How to get the name of browser using Web Driver?**

**public class JsExecute**

**{**

**WebDriver driver;**

**JavascriptExecutor js;**

**@Before**

**public void setUp() throws Exception**

**{**

**driver=new FirefoxDriver();**

**driver.get("http://www.google.com");**

**}**

**@Test**

**public void test()**

**{**

**JavascriptExecutor js = (JavascriptExecutor) driver;**

**System.out.println(js.executeScript("return navigator.appCodeName"));**

**}}**

**OR**

**String s = (String) ((JavascriptExecutor) driver).executeScript("return navigator.userAgent;");**

**System.out.println("Browser name : " + s);**

1. **How to handle colors in web driver?**

**Use getCssValue(arg0) function to get the colors by sending 'color' string as an argument.**

**Example**

**String col = driver.findElement(By.id(locator)).getCssValue("color");**

1. **How to pass parameters from testng.xml into  
   test case.**

**import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.chrome.ChromeDriver;  
import org.openqa.selenium.firefox.FirefoxDriver;  
import org.openqa.selenium.htmlunit.HtmlUnitDriver;  
import org.openqa.selenium.ie.InternetExplorerDriver;  
import org.testng.annotations.BeforeTest;  
import org.testng.annotations.Parameters;  
import org.testng.annotations.Test;  
  
public class Parallelexecution {  
  
private WebDriver driver = null;  
  
@BeforeTest  
@Parameters({ "BROWSER" })  
public void setup(String BROWSER) {  
System.out.println("Browser: " + BROWSER);  
  
if (BROWSER.equals("FF")) {  
System.out.println("Firefox Browser is selected");  
driver = new FirefoxDriver();  
} else if (BROWSER.equals("IE")) {  
System.out.println("Internet Explorer Browser is selected");  
driver = new InternetExplorerDriver();  
} else if (BROWSER.equals("HU")) {  
System.out.println("Html Unit Browser is selected");  
driver = new HtmlUnitDriver();  
} else if (BROWSER.equals("CH")) {  
System.out.println("Google chrome Browser is selected");  
driver = new ChromeDriver();  
}  
}  
  
@Test  
public void testParallel() throws Exception {  
driver.get("http://ruchi-myseleniumblog.blogspot.in/2013/12/100-selenium-interview-questions.html");  
  
}  
}**

1. **How to get text from captcha image??**

**driver.findElement(By.xpath(".//\*[@id='SkipCaptcha']")).click();**

**String attr = ie.findElement(By.xpath(".//\*[@id='SkipCaptcha']")).getAttribute("value");**

**System.out.println("The value of the attribute 'Name' is " + attr);**

1. **Is there a way to click hidden LINK in web driver?**

**String Block1 = driver.findElement(By.id("element ID"));**

**JavascriptExecutor js1=(JavascriptExecutor)driver;**

**js1.executeScript("$("+Block1+").css({'display':'block'});");**

1. **What Class Extends Web Driver?**

* **· AndroidDriver, ChromeDriver, EventFiringWebDriver, FirefoxDriver, HtmlUnitDriver,   
  InternetExplorerDriver, IPhoneDriver, PhantomJSDriver, RemoteWebDriver, SafariDriver**

1. **What are the APIs that support Web Driver?**

* **· API are nothing but collection of all selenium commands for Locating UI Elements   
  (WebElements),Fetching a Page,User Input etc…**

1. **How to disable cookies in browser.**

* **· Using deleteAllVisibleCookies() in selenium**

1. **"We have heard about frameworks well it can be  
   broadly classified into these TDD, BDD and ATDD frameworks .What’s the Difference?"**

**TDD- Test Driven Development, Behaviour Driven Development & Acceptance TestDriven Development**

**Well, you could see the above Acronyms buzzing over all Automation folks. I was not sure on what it means and How it differs each other. How each methodology will benefit? and where exactly it will help in the Development Life cycle.**

**Finally, after some analysis I had found out the differences and posting it here. Readers are always welcomed to correct me if I am wrong.**

***First lets list out what exactly each methodology does means***

**TDD – Test Driven Development**

**Its also called test-driven design, is a method of software development in which unit testing is repeatedly done on source code. Write your tests watch it fails and then refactor it. The concept is we write these tests to check if the code we wrote works fine. After each test, refactoring is done and then the same or a similar test is performed again. The process is iterated as many times as necessary until each unit is functionally working as expected. TDD was introduced first by XP. I believe I have explained enough in simple terms.**

**BDD – Behaviour Driven Development**

**Behavior-driven development combines the general techniques and principles of TDD with ideas from domain-driven design**

[***DDD-Domain Driven Testing***](http://domaindrivendesign.org/resources/what_is_ddd) **BDD is similar in many ways to TDD except that the word “test” is replaced with the word “Behaviour”. It’s purpose is to help the the folks devising the system (i.e., the developer) identify appropriate tests to write–that is, tests that reflect the behavior desired by the stakeholders. BDD is usually done in very English-like language helps the Domain experts to understand the implementation rather than exposing the code level tests. Its defined in a GWT format, GIVEN WHEN & THEN.**

1. **How to change user agent in Firefox by selenium  
   web driver.**

**FirefoxProfile profile = new FirefoxProfile();**

**profile.setPreference("general.useragent.override", "some UA string");**

**Web Driver driver = new FirefoxDriver(profile);**

1. **What is Selenese?**

**Selenese is HTML language based command, which is used in Selenium IDE.**

1. **Differences between QTP and selenium.**

**1) Selenium generates a proxy while starting browser. QTP does not**

**2) QTP uses only Vb script. Selenium is available in many languages**

**3) QTP is paid and selenium is free.**

**4) You can run script from a particular line in QTP but in selenium, you cannot.**

**5) Selenium works on all browsers. QTP only works on IE, mozilla. Support from chrome has been introduced lately.**

**6) QTP is more organized and user friendly**

**7) Selenium requires more technical skills**

**8) QTP can also be used on desktop based applications but selenium cannot be used**

1. **What is the MOST challenging test problem in my**

**career in Automation?**

**In my career**

* **· Changing XPATHS' between testing server and production server-by keeping generic xpath**
* **· Keep separate property files for production and UAT**
* **· automating flash apps**
* **· Mobile Automation**

1. **“Suppose developer changed the existing image to**

**new image with same xpath. Is test case pass or fail?"**

* **· Pass**

1. **How to handle network latency using selenium?**

* **· Using driver.manage.pageLoadingtime for network latency**

1. **How does u handle dynamic elements without using**

**xpath (with example?)**

* **· By using classname or css.**

1. **What are the different types of driver implementation?**

* **· AndroidDriver, AndroidWebDriver, ChromeDriver, EventFiringWebDriver, FirefoxDriver, HtmlUnitDriver, InternetExplorerDriver, IPhoneDriver, IPhoneSimulatorDriver, RemoteWebDriver, SafariDriver, WebDriverBackedSelenium**

1. **Code for Opening Firefox browser?**

* **· Webdriver driver=new FireFoxdriver();**

1. **Which repository you have used to store the test  
   scripts?**

**I have created scripts in excel file and store them in Test cases folder under src .**

1. **How to work with radio button in web driver?**

**We can select the value from the drop down by using 3 methods.**

**selectByVisibleText - select by the text displayed in drop down**

**selectByIndex - select by index of option in drop down**

**selectByValue - select by value of option in drop down**

**<select id="44"><option value="1">xyz</option>**

**<option value="2">abc</option>**

**<option value="3">pqr</option>**

**</select>**

**WebElement e = driver.findElement(By.id("44"));**

**Select selectElement=new Select(e);**

**// both of the below statements will select first option in the weblist**

**selectElement.selectByVisibleText("xyz");**

**selectElement.selectByValue("1");**

1. **How to work with dynamic web table?**

**You can get the total number of <tr> tags within a <td> tag by giving the xpath of the  
<td> element by using this function -**

**List<WebElement> ele = driver.findElements(By.xpath("Xpath of the table"));**

**Now you can use a for each loop to loop through each of the <tr> tags in the above list   
and then read each value by using getText() method.**

1. **Detail about TestNG Test Output folder.**

**It is the directory where reports are generated. Every time tests run in a suite, TestNG   
creates index.html and other files in the output directory.**

1. **In frame if no frame Id as well as no frame  
   name then which attribute I should consider   
   throughout our script.**

**You can go like this.....driver.findElements(By.xpath("//iframe"))...**

**Then it will return List of frames then switch to each and every frame and search for  
the locator which you want then break the loop**

1. **What is object repository?**

**It is collection of object names their properties, attributes and their values .It maye be  
excel, XML,property file or text file**

1. **TestNG vs. Junit?**

**Advantages of TestNG over Junit**

* **In Junit we have to declare @BeforeClass and @AfterClass which is a constraint where as in TestNG there is no constraint like this.**
* **Additional Levels of setUp/tearDown level are available in TestNG like @Before/AfterSuite,@Before/AfterTest and @Before/AfterGroup**
* **No Need to extend any class in TestNG.**
* **There is no method name constraint in TestNG as in Junit. You can give any name to the test methods in TestNG**
* **In TestNG we can tell the test that one method is dependent on another method where as in Junit this is not possible. In Junit each test is independent of another test.**
* **Grouping of testcases is available in TestNG where as the same is not available in Junit.**
* **Execution can be done based on Groups. For ex. If you have defined many cases and segregated them by defining 2 groups as Sanity and Regression. Then if you only want to execute the “Sanity” cases then just tell TestNG to execute the “Sanity” and TestNG will automatically execute the cases belonging to the “Sanity” group.**
* **Also using TestNG your selenium test case execution can be done in parallel.**

1. **What is the difference between @before   
   method and @beforeclass.**

**In JUnit4 @Before is used to execute set of preconditions before executing a test.   
For example, if there is a need to open some application and create a user before   
executing a test, then this annotation can be used for that method. Method that is   
marked with @Before will be executed before executing every test in the class.**

**If a JUnit test case class contains lot of tests which all together need a method   
which sets up a precondition and that needs to be executed before executing the   
Test Case class then we can utilise “@BeforeClass” annotation.**

1. **What are the different Parameters for @Test   
   annotation?**

**Parameters are keywords that modify the annotation’s function.**

1. **Can we run group of test cases using TestNG?**

**Test cases in group in Selenium using TestNG will be executed with the below options.**

**If you want to execute the test cases based on one of the group like regression test or smoke test**

**@Test(groups = {"regressiontest", "smoketest"})**

1. **Differences between Selenium web driver,   
   IDE and RC?**
2. **How to highlight an object like qtp/uft does   
   through selenium and java?**

**public void highlightElement(WebDriver driver, WebElement element) {**

**for (int i = 0; i < 2; i++)**

**{**

**JavascriptExecutor js = (JavascriptExecutor) driver;**

**js.executeScript("arguments[0].setAttribute('style', arguments[1]);", element, "color: yellow; border: 2px solid yellow;");**

**js.executeScript("arguments[0].setAttribute('style', arguments[1]);", element, "");**

**}}**

**Call the highlightElement method and pass webdriver and WebElement which you want to highlight as arguments.**

1. **What are the different assertions in SIDE?**

**Assertionsare like Accessors, but they verify that the state of the application conforms to what is expected. Examples include "make sure the page title is X" and "verify that this checkbox is checked".**

**All Selenium Assertions can be used in 3 modes: "assert", "verify", and "waitFor".  
  
For example, you can "assertText", "verifyText" and "waitForText". When an "assert" fails, the test is aborted. When a "verify" fails, the test will continue execution, logging the failure. This allows a single "assert" to ensure that the application is on the correct page, followed by a bunch of "verify" assertions to test form field values, labels, etc.  
  
"waitFor" commands wait for some condition to become true (which can be useful for testing Ajax applications). They will succeed immediately if the condition is already true. However, they will fail and halt the test if the condition does not become true within the current timeout setting (see the setTimeout action below).**

1. **How to store a value which is text box using   
   web driver?**

**driver.findElement(By.id("your Textbox")).sendKeys("your keyword");**

1. **How to handle alerts and confirmation boxes.**

**Confirmation boxes and Alerts are handled in same way in selenium.**

**var alert = driver.switchTo().alert();**

**alert.dismiss(); //Click Cancel or Close window operation**

**alert.accept(); //Click OK**

**Handle Confirmation boxes via JavaScript,**

**driver.executeScript("window.confirm = function(message){return true;};");**

1. **How to mouse hover on an element?**

**Actions action = new Actions(webdriver);**

**WebElement we = webdriver.findElement(By.xpath("html/body/div[13]/ul/li[4]/a"));**

**action.moveToElement(we).moveToElement(webdriver.findElement(By.xpath("/expression-here"))).click().build().perform();**

1. **How to switch between the windows?**

**private void handlingMultipleWindows(String windowTitle) {**

**Set<String> windows = driver.getWindowHandles();**

**for (String window : windows) {**

**driver.switchTo().window(window);**

**if (driver.getTitle().contains(windowTitle)) { return; } } }**

1. **How to switch between frames?**

**WebDriver's [driver.switchTo().frame()](http://selenium.googlecode.com/svn/trunk/docs/api/java/org/openqa/selenium/WebDriver.TargetLocator.html" \t "_blank) method takes one of the three possible arguments:**

* [**A number.**](http://selenium.googlecode.com/svn/trunk/docs/api/java/org/openqa/selenium/WebDriver.TargetLocator.html#frame%28int%29)

**Select a frame by its (zero-based) index. That is, if a page has three frames, the first frame would be at index "0", the second at index "1" and the third at index "2". Once the frame has been selected, all subsequent calls on the WebDriver interface are made to that frame.**

* [**A name or ID.**](http://selenium.googlecode.com/svn/trunk/docs/api/java/org/openqa/selenium/WebDriver.TargetLocator.html#frame%28java.lang.String%29)

**Select a frame by its name or ID. Frames located by matching name attributes are always given precedence over those matched by ID.**

* [**A previously found WebElement.**](http://selenium.googlecode.com/svn/trunk/docs/api/java/org/openqa/selenium/WebDriver.TargetLocator.html#frame%28org.openqa.selenium.WebElement%29)

**Select a frame using its previously located WebElement.**

**Get the frame by it's id/name or locate it by [driver.findElement()](http://selenium.googlecode.com/svn/trunk/docs/api/java/org/openqa/selenium/WebDriver.html" \l "findElement%28org.openqa.selenium.By%29" \t "_blank) and you'll be good.**

1. **What is actions class in web driver?**

**Actions class with web Driver help is Sliding element, Resizing an Element, Drag & Drop,**

**hovering a mouse, especially in a case when dealing with mouse over menus.**

**Dragging & Dropping an Element:**

**import org.openqa.selenium.By;**

**import org.openqa.selenium.WebDriver;**

**import org.openqa.selenium.WebElement;**

**import org.openqa.selenium.firefox.FirefoxDriver;**

**import org.openqa.selenium.interactions.Actions;**

**public class testDragandDrop {**

**public static void main(String[] args) throws InterruptedException {**

**WebDriver driver = new FirefoxDriver();**

**driver.get("http://jqueryui.com/resources/demos/droppable/default.html");**

**WebElement draggable = driver.findElement(By.xpath("//\*[@id='draggable']"));**

**WebElement droppable = driver.findElement(By.xpath("//\*[@id='droppable']"));**

**Actions action = new Actions(driver);**

**action.dragAndDrop(draggable, droppable).perform();**

**}**

**}**

**Sliding an Element:**

**import org.openqa.selenium.By;**

**import org.openqa.selenium.WebDriver;**

**import org.openqa.selenium.WebElement;**

**import org.openqa.selenium.firefox.FirefoxDriver;**

**import org.openqa.selenium.interactions.Actions;**

**public class testSlider {**

**/\*\***

**\* @param args**

**\* @throws InterruptedException**

**\*/**

**public static void main(String[] args) throws InterruptedException {**

**WebDriver driver = new FirefoxDriver();**

**driver.get("http://jqueryui.com/resources/demos/slider/default.html");**

**WebElement slider = driver.findElement(By.xpath("//\*[@id='slider']/a"));**

**Actions action = new Actions(driver);**

**Thread.sleep(3000);**

**action.dragAndDropBy(slider, 90, 0).perform();**

**}**

**}**

**Re-sizing an Element:**

**import org.openqa.selenium.By;**

**import org.openqa.selenium.WebDriver;**

**import org.openqa.selenium.WebElement;**

**import org.openqa.selenium.firefox.FirefoxDriver;**

**import org.openqa.selenium.interactions.Actions;**

**public class testResizable {**

**public static void main(String[] args) throws InterruptedException {**

**WebDriver driver = new FirefoxDriver();**

**driver.get("http://jqueryui.com/resources/demos/resizable/default.html");**

**WebElement resize = driver.findElement(By.xpath("//\*[@id='resizable']/div[3]"));**

**Actions action = new Actions(driver);**

**action.dragAndDropBy(resize, 400, 200).perform();**

**}**

**}**

1. **Difference between the selenium1.0 and   
   selenium 2.0?**

**Selenium 1 = Selenium Remote Control.**

**Selenium 2 = Selenium Web driver, which combines elements of Selenium 1 and Web driver.**

1. **Difference between find element () and   
   findelements ()?**

**findElement() :**

**Find the first element within the current page using the given "locating mechanism".**

**Returns a single WebElement.**

**findElements() :**

**Find all elements within the current page using the given "locating mechanism".**

**Returns List of Web Elements.**

1. **How to take the screen shots in seelnium2.0?**

**// store screenshots  
public static void captureScreenShot(String filePath) {  
File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);  
try {  
FileUtils.copyFile(scrFile, new File(filePath));  
} catch (IOException e) {  
// TODO Auto-generated catch block  
e.printStackTrace();  
  
}   
}**

1. **What is the default time for selenium Ide and   
   webdriver?**

**Default timeout in selenium ide is 30 seconds.**

1. **Write down scenarios which we can't automate?**

**Barcode Reader, Captcha etc.**

1. **In TestNG I have some test's Test1-Test2-  
   Test3-Test4-Test5I want to run my execution   
   order is Test5-Test1-Test3-Test2-Test4.How   
   do you set the execution order can you explain   
   for that?**

* **· Use priority parameter in @test annotation or TestNG annotations.**

1. **Differences between jxl and ApachePOI.**

* **· jxl does not support XLSX files**
* **· jxl exerts less load on memory as compared to ApachePOI**
* **· jxl doesn't support rich text formatting while ApachePOI does.**
* **· jxl has not been maintained properly while ApachePOI is more up to date.**
* **· Sample code on Apache POI is easily available as compare to jxl.**

1. **How to ZIP files in Selenium with an Example?**

**// Sample Function to make zip of reports  
public static void zip(String filepath){  
try  
{  
File inputFolder=new File('Mention file path her");  
File outputFolder=new File("Reports.zip");  
ZipOutputStream out = new ZipOutputStream(new BufferedOutputStream(new FileOutputStream(outputFolder)));  
BufferedInputStream in = null;  
byte[] data = new byte[1000];  
String files[] = inputFolder.list();  
for (int j=0; j<files.length; i++)  
{  
in = new BufferedInputStream(new FileInputStream  
(inputFolder.getPath() + "/" + files[j]), 1000);   
out.putNextEntry(new ZipEntry(files[j]));  
int totalcount;  
while((totalcount= in.read(data,0,1000)) != -1)  
{  
out.write(data, 0, totalcount);  
}  
out.closeEntry();  
}  
out.flush();  
out.close();   
}  
catch(Exception e)  
{  
e.printStackTrace();  
return "Fail - " + e.getMessage();  
}  
}**

1. **What is default port no?**

**4444**

1. **If Default port no is busy how to change port no?**

**We can use any port number which is valid.. First create an object to remote control configuration.   
Use 'setPort' method and provide valid port number(4545,5555,5655, etc).. There after attach this   
remote control configuration object to selenium server..i.e**

**RemoteControlConfiguration r= new RemoteControlConfiguration();**

**r.setPort(4567);**

**SeleniumServer s= new SeleniumServer(r);**

1. **Does Selenium support https protocols?**

**Yes**

1. **Majorly asked test scenario with framework in   
   Interviews?**

**Majorly asked are:**

* **Login for Gmail scenario**
* **Goggle search and finding no of results**
* **Downloading a file and save it**
* **Checking mails and deleting them**
* **Do shopping in flipkart.com**

1. **Selenium support mobile applications?**

**No, it is browser automation tool, it only automates Websites opening in mobile browser, and mobile APPs**

**can't be automated.**

1. **What is wraps Driver?**

**For casting selenium instance to selenium2 (webdriver). wraps driver is used.**

**For more details.**

1. **Can you explain Junit Annotation? If there are   
   1000 test cases. 500 test cases are executed. How  
   will you execute the rest of the test cases by using annotation?"**

**The annotations generated with JUnit 4 tests in Selenium are:**

1. **@Before public void method() - Will perform the method() before each test. This method  
   can prepare the test**
2. **@Test public void method() - Annotation @Test identifies that this method is a test   
   method.environment,e.g. read input data, initialize the class)**
3. **@After public void method() - Test method must start with test@Before - this annotation  
   is used for executing a method before**

1. **Difference between assert and verify in selenium  
   web driver.**

* **· When an “assert” fails, the test will be aborted. Assert is best used when the  
  check value has to pass for the test to be able to continue to run log in.**
* **· Where if a “verify” fails, the test will continue executing and logging the failure.   
  Verify is best used to check non critical things. Like the presence of a  
  headline element.**

1. **"I want to find the location of ""b"" in the below  
   code, how can I find out without using xpath, name,  
   id, csslocator, index.<div>**

**<Button>a</button>**

**<Button>b</button>**

**<Button>c</button>**

**</div>  
· driver.findElement(By.xpath("//\*[contains(text(),'b')]")).click(); or**

* **//div/button[contains(text(),'b']**
* **·**

1. **How to do Applet testing using selenium?**

**// selenium setup  
selenium = new DefaultJavaSelenium("localhost",4444, browserString , url);  
selenium.start();  
selenium.open(url);  
  
// get the appletfixure to control fest JAppletFixture  
AppletFixture dialog = selenium.applet(LIST\_APPLET\_ID)  
  
// fest similar API for autmation testing  
dialog.comboBox("domain").select("Users");  
dialog.textBox("username").enterText("alex.ruiz");  
dialog.button("ok").click();**

1. **Name 5 different exceptions you had in   
   selenium web driver and mention what instance   
   you got it and how do you resolve it?**

* **· WebDriverException**
* **· NoAlertPresentException**
* **· NoSuchWindowException**
* **· NoSuchElementException**
* **· TimeoutException**

1. **How do you manage the code versions in   
   your project?**

* **· Using SVN or other versioning tools**

1. **Latest version of Firefox and selenium in   
   market and the version on which you are testing  
   which you are testing.**

* **· FF Latest version till Dec,2013 for windows7,64 bit :26.0.I use FF 25.0.1 (ur ans. may differ)**
* **· Selenium web driver latest version till dec,2013- 2.39.0 I use selenium 2.37 see latest at**

1. **How to know all the methods supported in   
   web driver**

**and its syntax.**

* **· In Org.openqa.selenium package, web driver interface has all the main methods that can**

**be used in Selenium Web driver**

* **·**

1. **How do you create html test report from   
   your test script?**

* **I would see below 3 ways:**
* **Junit: with the help of ANT.**
* **TestNG: using inbuilt default.html to get the HTML report. Also XLST reports from ANT,**

**Selenium, TestNG combination.**

* **Using our own customized reports using XSL jar for converting XML content to HTML.**

1. **List the browsers, OS supported by the Selenium**

**Windows Linux Mac**

**IE Y NA NA**

**FF Y Y Y**

**Safari Y N Y**

**Opera Y Y Y**

**Chrome Y Y Y**

1. **Can you explain Selenium Mobile Automation?**

**import junit.framework.TestCase;  
  
import org.openqa.selenium.By;  
import org.openqa.selenium.WebElement;  
import org.openqa.selenium.android.AndroidDriver;  
  
public class OneTest extends TestCase {  
  
public void testGoogle() throws Exception {  
WebDriver driver = new AndroidDriver();  
  
// And now use this to visit Google  
driver.get("http://www.google.com");  
  
// Find the text input element by its name  
WebElement element = driver.findElement(By.name("q"));  
  
// Enter something to search for  
element.sendKeys("Cheese!");  
  
// Now submit the form. WebDriver will find the form for us from the element  
element.submit();  
  
// Check the title of the page  
System.out.println("Page title is: " + driver.getTitle());  
driver.quit();  
}  
}**

1. **What mobile devices it may Support?**

**Selenium Web driver supports all the mobile devices operating on Android, IOS operating Systems**

* **· Android – for phones and tablets (devices & emulators)**
* **· iOS for phones (devices & emulators) and for tablets (devices & emulators)**

1. **What is the difference between single and  
   double slash**

**in Xpath?**

**/**

**1.It starts selection from the document node**

1. **It Allows you to create 'absolute' path expressions**
2. **e.g “/html/body/p” matches all the paragraph elements**

**//**

1. **It starts selection matching anywhere in the document**
2. **It Allows you to create 'relative' path expressions**
3. **e.g“//p” matches all the paragraph elements**

1. **What are the test types supported by Selenium?**

**Selenium supports UI and functional testing. As well it can support performance testing**

**for reasonable load using selenium grid.**

1. **In what all case we have to go for   
   “JavaScript executor”.**

**Consider FB main page after you login. When u scrolls down, the updates get loaded. To  
handle this activity, there is no selenium command. So you can go for javascript to set   
the scroll down value like driver.executeScript("window.scrollBy(0,200)", "");**