1. Java script executor :

http://www.guru99.com/execute-

javascript-selenium-webdriver.html

1. **What is the difference between type() and typeKeys() in WebDriver?**

type(locator,value)--Inputs the value. Sets the textfield to the value, nothing special. It just takes the entire string and puts it in there at one time.]

typekeys(locator,value)--Simulates each keystroke. So if the page has an action that occurs as you type (for example, google's autosuggest), this will be the action you would need.

1. **To click on specific part of element,** you would need to use clickAT command. ClickAt command accepts element locator and x, y co-ordinates as arguments- clickAt (locator, cordString)
2. **setSpeed** : Set execution speed (i.e., set the millisecond length of a delay which will follow each selenium operation). By default, there is no such delay, i.e., the delay is 0 milliseconds.

**Thread.sleep** : It causes the current thread to suspend execution for a specified period.So the main difference between them is setSpeed sets a speed while will apply delay time before every selenium operation takes place. But one thread.sleep() will set up wait only for once. So, if we have 3 selenium operations written like below:

Opeartion 1

Opeartion 2

Opeartion 3

and we want to set a delay time 2000 for each of these, defining setSpeed() method once will accomplish the task something like below:

selenium.setSpeed("2000");

Opeartion 1

Opeartion 2

Opeartion 3

But if we use Thread.sleep(), it will be something like below:

Thread.sleep(2000);

Opeartion 1

Thread.sleep(2000);

Opeartion 2

Thread.sleep(2000);

Opeartion 3

4. **How to Run failed test cases in Testng?**

We don't have to perform any other activity once you will get **testng**-**failed**.xml double click on this and analyze which **test case** are **failing** and why . Then modify your script and **run** it. To **run** above xml simple right click on xml then Select **run** as then **TestNG** Suite.

1. **Keyword Driven Framework** is a type of Functional Automation Testing **Framework** which is also known as Table-**Driven** testing or Action Word based testing.

 The basic working of the Keyword Driven Framework is to divide the Test Case in to four different parts.

**Test Step:** It is a very small description of the Test Step or the description of theAction going to perform on Test Object.  
**Test Object:** It is the name of the Web Page object/element, like Username & Password.  
**Action:** It is the name of the action, which is going to perform on any Object such as click, open browser, input etc.  
**Test Data:** Data can be any value which is needed by the Object to perform any action, like Username value for Username field.

1. **To perform an action when the test fails:**

Put it in the After method.

1. **@AfterMethod**
2. public void afterMethod() {
3. result = Reporter.getCurrentTestResult();
4. if (result.getStatus() == ITestResult.FAILURE) {
5. //your screenshooting code goes here
6. }
7. }

7. TestNG tutorials:

<https://www.tutorialspoint.com/testng/index.htm>

**Why use static methods than instance variables, what are the runtime advantages**

8. **Non-static** variable also known as instance variable while because memory is allocated whenever instance is created.

Non-static variable are specific to an object

Non-static variable can access with object reference.

**Static:** Memory is allocated at the time of loading of class so that these are also known as class variable.

Static variable are common for every object that means there memory location can be sharable by every object reference or same class.

Difference between java 7 and java8

Serialization and deserialization

Why can’t you instantiate abstract class

Difference between selenium server and hub

Multiple inheritance is not supported why?

Can a static method be overridden?

Reverse iterator

**Can an interface have default method?**

Default Methods In Java

Before Java 8, interfaces could have only abstract methods. The implementation of these methods has to be provided in a separate class. So, if a new method is to be added in an interface then its implementation code has to be provided in the class implementing the same interface. To overcome this issue, Java 8 has introduced the concept of default methods which allow the interfaces to have methods with implementation without affecting the classes that implement the interface.

// A simple program to Test Interface default

// methods in java

interface TestInterface

{

// abstract method

public void square(int a);

// default method

default void show()

{

System.out.println("Default Method Executed");

}

}

## **ENTRY CRITERIA**

Entry Criteria for QA testing is defined as “Specific conditions or on-going activities that must be present before a process can begin”. In the Systems Development Life Cycle it also specifies which entry criteria are required at each phase.

The type of required input from development includes:

1. Technical Requirements/Statement of Need
2. Design Document
3. Change Control
4. Turnover Document

The type of required input from test includes:

1. Evaluation of available software test tools
2. Test Strategy
3. Test Plan
4. Test Incident Reports.

**Exit Criteria**

is often viewed as a single document commemorating the end of a life cycle phase. Exit Criteria is defined as “The specific conditions or on-going activities that must be present before a life cycle phase can be considered complete.

The type of output from test includes:

1. Test Strategy
2. Test Plan
3. Test Scripts/Test Case Specifications
4. Test Logs
5. Test Incident Report Log
6. Test Summary Report/Findings Report

**Test strategy** is a high level document which defines the **approach for software testing**. It is basically derived from the **Business Requirement document**.( *The emphasis in a BRD is on what is required, rather than on how to achieve it,* *type of project, the needs and preferences of your business and technical stakeholders*,*)*

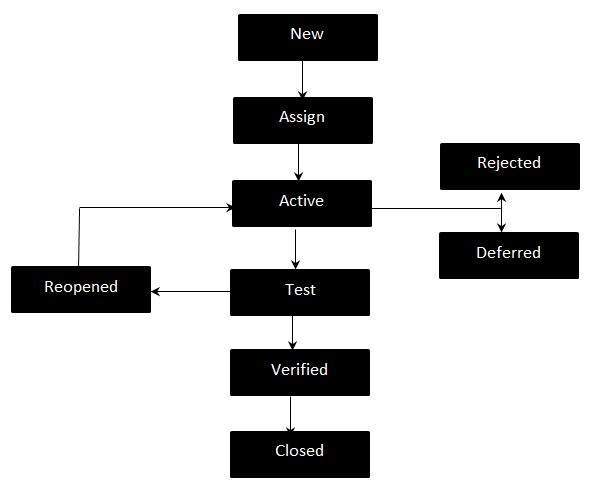
Test strategy is developed by project manager or business analyst. It is kind of static document which sets the standards for testing so not updated often.

* 1. **Scope and objective**
  2. **Business Issues:** How much is the budget of the project, how much time is required for testing, how much resources are needed etc. are the part of business issues which needs to be considered before the actual testing starts.
  3. **Testing approach:** What type of testing is needed
  4. **Test deliverables:** What are the documents required from the testing team,
  5. **Defect tracking approach**:
  6. **Training**
  7. **Automation**:
  8. **Risks**

**Test plan** is derived from **SRS (Software Requirement Specification)** which is prepared by test lead or manager. The main goal of test plan is to include all the details related to testing such as **what to test, when to test, how to test and who will be the tester.** Test plan is often not updated but if there is some new feature or change is introduced then it has to be updated accordingly

1. **Test plan ID**
2. **Test environment:**
3. **Features to be tested/Not tested:**
4. **Entry/Exit criteria:**
5. **Status:**
6. **Types of testing:**

## **Defect Life Cycle - Workflow:**



**Deferred -**When a defect cannot be addressed in that particular cycle it is deferred to future release.

**Rejected -**A defect can be rejected for any of the 3 reasons; viz - duplicate defect, NOT a Defect, Non Reproducible.

**Severity** of a defect is related to **how severe a bug is**. Usually the severity is defined in terms of **financial loss, damage to environment**, company’s reputation and loss of life.

**Priority** of a defect is related to **how quickly a bug should be fixed** and deployed to **live servers**. When a defect is of high severity, most likely it will also have a high priority. Likewise, a low severity defect will normally have a low priority as well.

### **High Severity – High Priority bug**

Every customers get error message on the booking form

### **High Severity – Low Priority bug**

Customers who use very old browsers cannot continue with their purchase of a product. Because the number of customers with very old browsers is very low, it is not a high priority to fix the issue.

### **High Priority – Low Severity bug**

the logo or name of the company is not displayed on the website.

### **Low Priority – Low Severity bug**

he privacy policy page take a long time to load. Not many people view the privacy policy page and slow loading doesn’t affect the customers much.

# Difference between software Verification and Validation:

|  |  |
| --- | --- |
| **Verification** | **Validation** |
| **Are we building the system right?** | **Are we building the right system?** |
| **Verification** is the process of evaluating products of a development phase to find out whether they meet the specified requirements. | **Validation** is the process of evaluating software at the end of the development process to determine whether software meets the customer expectations and requirements. |
| The objective of Verification is to make sure that the product being develop is as per the requirements and design specifications. | The objective of Validation is to make sure that the product actually meet up the user’s requirements, and check whether the specifications were correct in the first place. |
| Following activities are involved in **Verification**: Reviews, Meetings and Inspections. | Following activities are involved in **Validation**: Testing like black box testing, white box testing, gray box testing etc. |
| **Verification** is carried out by QA team to check whether implementation software is as per specification document or not. | **Validation** is carried out by testing team. |

# Requirements Traceability Matrix

1. The **Requirements Traceability Matrix** (RTM) is a document that links requirements throughout the validation process
2. The purpose of the Requirements Traceability Matrix is to ensure that all requirements defined for a system are tested
3. In the example shown below, requirements are traced between a Functional Requirements Specification, Design Specification, and Operational Qualification.

|  |  |  |
| --- | --- | --- |
| **Functional Requirements** | **Design Specifications** | **Test Cases** |
| The program will have a functional audit trail. | Each form will use fxn\_Audit\_Trail in the OnUpdate event procedure. | OQ, Test Case 3, Step 52: Audit Trail Verification |