Selenium – 3.142

Eclipse – 4.20

Maven – 3.8.1

TestNG – 7.1.0

Core java – 8

Extent reports – 4

Mysql workbench 8.0

Clindoc → Manual, automate UI

User CP → Manual, automation, functional and UI

Patient registration - Manual, automation, functional and UI

Maa-Hindi – Manual, automation.

Selenium 3 vs 4
https://www.softwaretestinghelp.com/new-features-in-selenium-4/

Revision -

Selenium(screenshot, broken links, DefaultSelenium class, desired Capabilities, selenium 4, locators, write xpath/css without using indexes, traverse to sibling element, traverse from paret-child and child-parent, relative vs absolute, alerts and popups, handle child windows, iframes, chromeOptions, handle session and cookie,)/selenium components(webdriver architecture, RC, Grid, IDE)/TestNG(annotations, imp of .xml file, helper attributes, hard vs soft assert, ITestListener, @findBy)/Maven(folder structure, why maven, surefire plugin, artifact and groupId, ant vs maven, features, maven build lifecycle, types of maven repo)/POM(what is POM, @findBy)/cucumber/Jenkins(what is Jenkins, common test/use cases jenkins used for, what is Jenkins job, what is continuous integration, advantages, Jenkins pipeline, how to store Jenkins credentials securely, different ways to trigger Jenkins job, process of Jenkins, pre-requisites of using jenkins)/git/log4j(what is log4j, logger class)/extent reports/manual testing(regression, retesting, ad-hoc, sanity, smoke, unit testing, what is test plan, test scripts, test cases, traceability matrix, bug life cycle, SDLC, STLC) /excel(read data from excel and print it)/challenges u faced with selenium/SSL certificate/core java/programs/which framework u used in your previous company/common exceptions

Revision –

How you assigned bug to developer

Watch video of any bug reporting tool

Prepare intro

Explain project

Roles and responsibility

Bdd vs tdd

- Contract between hashcode and equals

The contract between equals() and hashCode() is: 1) **If two objects are equal, then they must have the same hash code**. 2) If two objects have the same hash code, they may or may not be equal. ... The default implementation of hashCode() in Object class returns distinct integers for different objects.

1. Go through all projects which you created while doing udemy classes
2. read data from excel and write to application prog(pg no-109 naresh it)
3. take screenshot on failure (pg-110 naresh it)
4. take screenshot of failed test case/success test case
5. drag n drop(pg-111 naresh it)

6. How to pass unique id and password every time through automation(use timestamp with date)…google it
7. What are technical challenges you have faced in the automation project and how did you overcome those
8. If you were given a chance to develop a framework what framework you will be choose(BDD(Cucumber), POM) why?
9. Handle popup whether it is displayed or not(cz sometimes popup is displayed and sometimes its not displayed) (lecture - 315) (if size > 0 then element is present) refer links.docx Q.38
10. Locate svg elements
11. File download
12. In soft assert if test fails even though it will continue to execute, but will it show error ? ➔ yes it will show AssertionError
13. If u have given choice to use assert/soft assert what will u choose and why?
14. Methods of assert
15. How POM works? ➔ https://www.guru99.com/page-object-model-pom-page-factory-in-selenium-ultimate-guide.html

    in POM there will be a separate class of each page and each page contains locators(here locators can be private to achieve encapsulation) and methods which define action needs to perform on those elements. And then will create object of these pages and can use these methods.
16. What are rules needs to be followed in method overriding and method overloading

    https://www.javatpoint.com/method-overloading-vs-method-overriding-in-java
17. What is end-end testing
18. what is StaleReferenceException in POM
19. integration testing example

20. test scenario, test case, test script, test plan

    https://www.guru99.com/test-case-vs-test-scenario.html

21.

Mock interview

when you compile program --> it is converted into .class file which is result of compilation process

-->once it goes from interpretation then it is converted in binary

--> cz computer only knows 0's and 1's

Mock interview

- locbating elements
- Web element is class/interface and its methods
- what will happen when you try to perform action on enabled and disabled web element
- click on link without using click method => keys control + enter

=> String clickOnLink = Keys.chord(Keys.CONTROL, Keys.ENTER);
=>firstColumn.findElements(By.tagName("a")).get(i).sendKeys(clickOnLink);

- enter username -> copy same username and paste it in password field how to do that

=>ctrl + A , then ctrl + c in username field

=>ctrl + v in password field

- driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS); => correct syntax

- dropdowns => static and dynamic

  => select and deselect... check DeselectOptionExample in introduction

  => isMultiple isMultiple( ) command is used to verify whether the specified select element support selecting multiple options at the same time

- what is debugging and when you use it => e.g. if some link is not working then will check application logs
- Application log = > an application log is a file that contains information about events that have occurred within a software application. These events are logged out by the application and written to the file. They can include errors and warnings as well as informational events.

- Whys synchronization come in picture => In the Multithreading concept, multiple threads try to access the shared resources at a time to produce inconsistent results. The synchronization is necessary for reliable communication between threads

- How synchronization is achieve in selenium=>using implicit/explicit waits

Exceptions which u came across while working-

1. NoSuchElementException - This Exception occurs if an element could not be found
2. NoSuchFrameException - This Exception occurs if the frame target to be switched to does not exist.
3. TimeoutException - the element searched wasn't found in the specified time.
4. NoAlertPresentException - This Exception occurs when you switch to no presented alert.
5. NoSuchWindowException - This Exception occurs if the window target to be switch does not exist.
6. `ElementClickInterceptedException`

Basic selenium Questions:

1. What is selenium? What Is selenium grid/RC/webdriver?

⇨ **Selenium** → It is Open source web automation tool.

→selenium can be used to automate functional tests and can be integrated with Maven, Jenkins. It is also integrated with TestNG to generate reports.

→It supports automation across different browsers, platforms and programming lang

→It supports for mobile applications like iOS and android using Appium.

→Lang supported by selenium include C#, java, perl, php, python

- ⇨ Selenium Grid, Selenium RC, Selenium Webdriver, IDE → https://www.edureka.co/community/2477/difference-between-selenium-ide-rc-and-webdriver

2. Difference between get and navigate method in selenium?

   Ans: get() is used to navigate particular URL(website) and wait till page load. And when we try to interact with element before it completely loads complete webpage it will throw an error

   navigate() is used to navigate to particular URL and does not wait to page load. This will hit url and will continue execution of next steps.(then what is use of creating navigate method- there are some methods present like back, forward/refresh same is not achieved using get())

3. Quit() vs close()

   close – It closes the the browser window on which the focus is set(suppose there are 3 browsers open and focus is on 2nd then 2nd browser will b closed)

   quit – all the browsers r closed

   dispose method which in turn closes all the browser windows and ends the WebDriver session gracefully.

4. Impicit vs explicit wait

   Implicit waits - are used to provide a default waiting time between each consecutive test step/command across the entire test script

   ```
   driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);
   ```

Explicit - we declared implicit Wait of **10 secs** but an element takes more than that, say 20 seconds and sometimes may appears on 5 secs, so in this scenario, Explicit wait is declared

5. Different ways to handle frames in application using web driver methods

   Ans: driver.swichTo().frame(index/name(or id)/frame webelement);

   Three ways by index/name(or id)/frame webelement

6. How to handle 3rd child window?

Ans:
```
Set<string> s = driver.getWindowHandles();     //ref of all opened browsers
    Iterator it = s.iterator();                //iterate through each browser
    It.next(); //0th index                     //0th
    It.next(); //1st index                     //1st
    String thirdId = It.next(); //2nd index    //2nd
      Driver.switchTo.window(thirdId);         //now switch to 3rd window
```

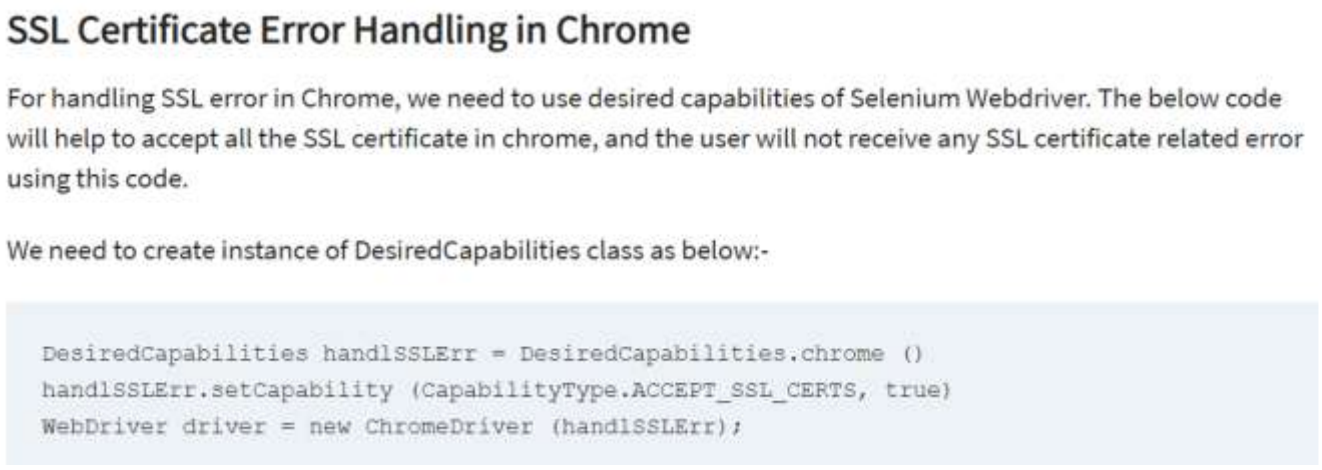
In simpler terms, driver. getWindowHandles() stores the set of handles for all the pages opened simultaneously, but driver. getWindowHandle() fetches the handle of the web page which is in focus. It gets the address of the active browser and it has a return type of String

driver.getWindowHandle(); //is to get reference object(informally handle) for the current browser window. (which is rarely useful)

vs

driver.getWindowHandles(); //is to get reference objects for the all opened browser windows

7. Handle https certifications:



SSL Certificate Error Handling in Chrome

For handling SSL error in Chrome, we need to use desired capabilities of Selenium Webdriver. The below code will help to accept all the SSL certificate in chrome, and the user will not receive any SSL certificate related error using this code.

We need to create instance of DesiredCapabilities class as below:-

```
DesiredCapabilities handlSSLErr = DesiredCapabilities.chrome ()
handlSSLErr.setCapability (CapabilityType.ACCEPT_SSL_CERTS, true)
WebDriver driver = new ChromeDriver (handlSSLErr);
```

8. Types of locators in web driver
   Xpath/cssSelector/id/name/className/linkText/partialText/tagName

9. Syntax for xpath and css if id and tag are given
   Xpath = //tagName[@id='value']
   Css = tagName[id='value']

10. Xpath and css to use regular expression (using contains)
    We use regular expression xpath when id changes evry time when we refresh page e.g. id= username111, userName222…(here userName is common/remains same)

    //tagName[contains(@id,'userName')]   - xpath

    tagName[id*='userName']

11. Absolute vs relative xpath
    Absolute xpath
    - it starts with single slash
    - starts from html/body to actual element
    - element identification is faster compared to relative path
    - even if there is slight change in HTML DOM structure, absolute path will fail.

Relative xpath

- It starts with double slash
- This is more like starting simply by referencing the element you want and go from the particular location.
- Element identification is slower as compared to absolute path.
- Even if HTML DOM structure changes it will not affect relative path

12. https://www.youtube.com/watch?v=5tuiK5WS86s

13. https://www.youtube.com/watch?v=co4PO7t5aX4&list=PLFh6J1ZHhqNjC-ODipd9CVZWtX4uq7P3I

14. getOptions() -
```
Select optionSelect = new Select(driver.findElement(By.id("dropdown_cities")));

List <WebElement> elementCount = optionSelect.getOptions();

System.out.println(elementCount.size());
```

getOptions() is **used to get all options from the dropdown list**. This gets the all options belonging to the Select tag. It takes no parameter and returns ***List<WebElements>***

15.
```
if(selectElement.isMultiple())
{
System.out.println("Select tag allows multiple selection");
}
else
```

```
{
System.out.println("Select does not allow multiple selections");
}
```

isMultiple() method is **used to check whether the SELECT element support multiple selecting options** at the same time or not. This accepts nothing but returns boolean value(true/false).

16. getFirstSelectedOption() - We can get a selected option in a dropdown in Selenium webdriver. The method getFirstSelectedOption**()** returns the selected option in the dropdown. Once the option is fetched we can apply getText() method to fetch the text. And **return webElement**

TestNG:

1. What is TestNG
   - ⇨ TestNG is automation testing framework.
   - ⇨ We can generate reports using testNG(test-output folder >> emailable-report.html).
   - ⇨ you can easily come to know how many test cases are passed, failed, and skipped.
   - ⇨ You can execute the failed test cases separately(Once the execution is done, automatically test-output folder will be created in the project folder. In the test-output folder >> testng-failed.xml file will be created. Right click on testng-failed.xml>> Run as TestNg. This will now only re-run the fail testcase(s)).
   - ⇨ The same test case can be executed multiple times without loops just by using keyword called 'invocation count.

2. Adv and disadv of testNG, features
   - ⇨ Adv → parallel testing is possible
     →annotations r easy to understand
     →allows to assign priority to test methods
     →test cases can be grouped easily

→we can pass parameters via .xml to test method

- ⇨ Disdv
- ⇨ Features → adv

3. Annotations, @listener, @factory, @DataProvider, @BeforeTest vs @BeforeMethod
   - ⇨ @BeforeSuite – set env variables
   - ⇨ @BeforeTest
   - ⇨ @BeforeClass – open browser, obj creation, url
   - ⇨ @BeforeMethod
   - ⇨ @Test
   - ⇨ @AfterMethod
   - ⇨ @AfterClass – browser close
   - ⇨ @AfterTest
   - ⇨ @AfterSuite
   - ⇨ @BeforeGroups
   - ⇨ @AfterGroups
   - ⇨ @listeners → define listeners on test class. It is **an interface that listens to predefined events in test scripts and modifies the default behavior of the TestNG tool**
   - ⇨ @factory → annoted method allows tests to be created at runtime depending on certain data-sets/conditions. [tutorialspoint]
   - ⇨ @findBy →

4. Importance of testng.xml file
   - ⇨ When we test a project using Selenium Webdriver, it has a lot of classes on it. We cannot choose these classes one by one and put them for automation. Hence we need to create a suite so that all the classes run in a single test suite.
   - ⇨ We can achieve this by creating a testng.xml file
   - ⇨ It allows testers to create and handle multiple test classes, define test suites and tests.
   - ⇨ It allow execution of all test cases and run it under one xml file.
   - ⇨ It also provides parallel execution of test methods.
   - ⇨ testng.xml file allows to include or exclude the execution of test methods and test groups

- ⇨ It allows to pass parameters to the test cases

5. Pass parameter via .xml file to testcase.java
   - ⇨ In .xml file → `<parameter name="URL" value="qaacademy.com"/>`
   - ⇨ in test case →`@Parameters({"URL","APIKey/username"})`

6. List some common selenium assertions
   - ⇨ https://www.javatpoint.com/selenium-assertions
   - ⇨ **Assertion** determines the state of the application whether it is the same what we are expecting or not. If the assertion fails, then the test case is failed and stops the execution.
   - ⇨ **Hard Assertion** - Hard Assertion is an Assertion that throws the AssertException when the test case is failed. In the case of Hard Assertion, you can handle the error by using a catch block like a java exception. Suppose we have two test cases in a suite. The first test case in a suite has an assertion that fails, and if we want to run the second case in a suit, then we need to handle the assertion error
     - ➔ **Hard assertion contains following methods**
     - ➔ AssertFalse - Assertion verifies the boolean value returned by a condition. If the boolean value is false, then assertion passes the test case, and if the boolean value is true, then assertion aborts the test case by an exception

       Syntax

       Assert.AssertFalse(condition);

     - ➔ AssertTrue - Assertion verifies the boolean value returned by a condition. If the boolean value is true, then assertion passes the test case, and if the boolean value is false, then assertion aborts the test case by an exception

       Synatx

       Assert.AssertTrue(condition);

     - ➔ AssertEquals - AssertEquals() is a method used to compare the actual and expected results. If both the actual and expected results are same, then the assertion pass with no exception and

the test case is marked as "passed". If both the actual and expected results are not the same, then the assertion fails with an exception and the test case is marked as "failed"
Synatx –
Assert.assertEquals(actual,expected);

➔ AssertNotEquals – Opposite to function AssertEquals. Compare both expected and actual result, if not same then test will pass else test will fail.
Synatax
AssertNotEquals(actual,expected,message);

➔ AssertNull - AssertNull() is a method that verifies whether the object is null or not. If an object is null, then assertion passes the test case, and the test case is marked as "passed", and if an object is not null, then assertion aborts the test case, and the test case is marked as "failed”
Synatx –
Assert.assertNull(object);

➔ AssertNotNull - AssertNotNull() is a method that verifies whether the object is null or not. If an object is not null, then assertion passes the test case and test case is marked as "passed", and if an object is null, then assertion aborts the test case and test case is marked as "failed"
Syntax –
Assert.assertNotNull(object);

Note-In hard assertion, if an assertion fails then it aborts the test case otherwise it continues the execution. Sometimes we want to execute the whole script even if the assertion fails. This is not possible in Hard Assertion. To overcome this problem, we need to use a soft assertion in testng.

7. soft assert vs hard assert

8. Set priority to test case and if priority is not set what should be order of execution of test cases(what if -ve priority is set)
   - ⇨ If we don't mention any priority, testng will execute the @Test methods based on alphabetical order of their method names irrespective of their place of implementation in the code.
   - ⇨ By default priority is 0
   - ⇨ If same priority then by alphabetical order
   - ⇨ If -ve priority is given it consider it as 0 and if same priority is present then by alphabetical order

9. What is parameterized testng
   - ⇨ **Parameterization in Selenium** is a process to parameterize the test scripts in order to pass multiple data to the application at runtime
   - ⇨ There are two ways to pass parameters in test function parameters and dataProviders
   - ⇨ *the dataproviders can be run directly through the test case file* and we can't run parameters directly through test case file

10. How to create data driven framework i.e. @DataProvider
    - ⇨ The *DataProviders* in TestNG are another way to pass the parameters in the test function, the other one being TestNG parameters

11. Run group of test cases using testNG
    - ⇨ in test case @Test(groups= {"Smoke"})
    - ⇨ In .xml file

```
⇨ <test thread-count="5" name="Regression">
⇨           <groups>
⇨                   <run>
⇨                           <include name="Smoke"/>
⇨                   </run>
⇨           </groups>
⇨         <classes>
⇨         <class name="test.day3"/>
⇨         <class name="test.day1"/>
⇨         <class name="test.day2"/>
⇨      </classes>
⇨   </test>
```

12. Run test cases/classes In parallel

- ⇨ `<suite name="loan dept" parallel="tests" thread-count="2">`
- ⇨ `<suite name="loan dept" parallel="classes" thread-count="2">`
- ⇨ `<suite name="loan dept" parallel="methods" thread-count="2">`
- ⇨

13. Include/exclude particular test case/group from execution

⇨
```
      <class name="test.day3">
          <methods>
                    <exclude name="mobile.*"/>
           </methods>
        </class>
```

⇨
```
<class name="test.day3">
          <methods>
                    <include name="smoke"/>
           </methods>
        </class>
```

14. Disable test case in testNG

⇨ `@Test(enabled=false)`

15. Skip @test method

⇨ `@Test(enabled=false)`

16. Dependencies in testNG

- ⇨ Sometimes, you may need to invoke methods in a test case in a certain order. Here comes TestNG Dependencies into the picture.
- ⇨
```
@Test (dependsOnMethods = { "OpenBrowser" })
public void SignIn() {
System.out.println("This will execute second (SignIn)");}
```

17. Different ways to generate reports for testNG results (listeners/reporters)

- ⇨ test-output folder >> emailable-report.html
- ⇨ test-output folder >>index.html >>open with web browser

- ⇨ test-output folder >> testing-results.xml →xml report

18. How to write regular expression In testng.xml file to search @Test methods containing “smoke” keyword
    - ⇨ <methods>
        <exclude name=".*mobile.*"/>
      </methods>)

19. Use of invocationCount, enabled, priority, dependsOnMethod, dependsOnGroups, timeout,
    - ⇨ By using of invocationCount we can execute a test multiple times. Invocation count **determines how many times test will execute**
    - ⇨ Enabled = false → to disable test case
    - ⇨ Timeout → The timeOut is a helper attribute in TestNG that can **put an end to the execution of a test method if that method takes time beyond the timeOut duration**. A timeOut time is set in milliseconds, after that the test method will be marked Failed
    - ⇨

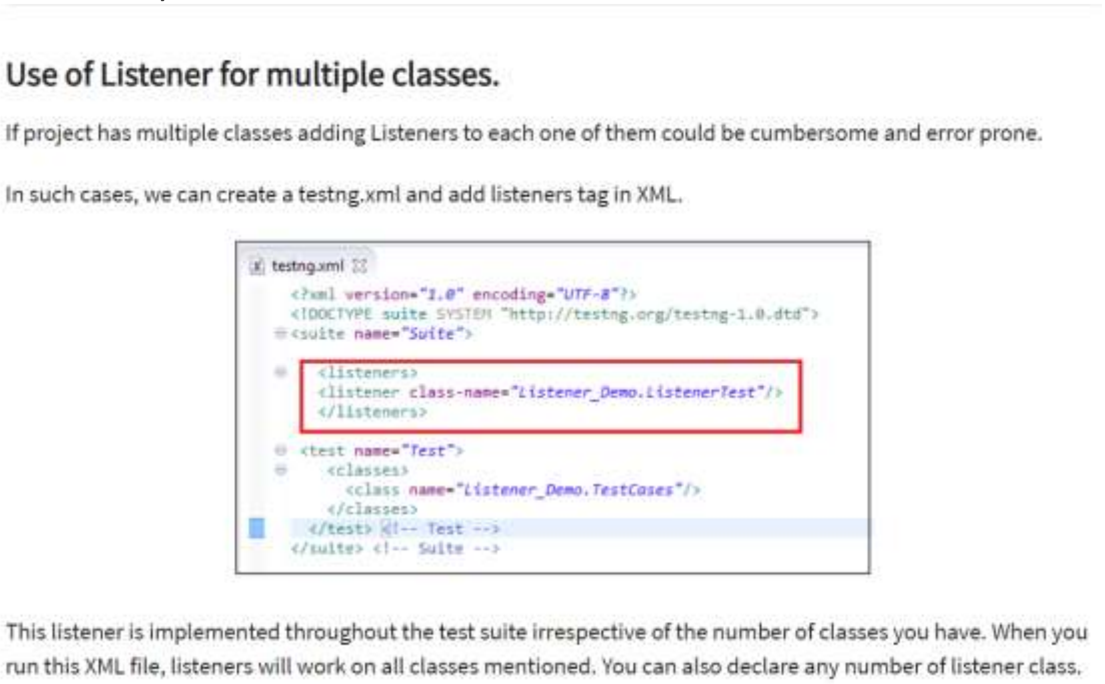
20. TestNG xml structure
    - ⇨ Suite – a suite contains one or more test elements
    - ⇨ Test – a test contains 1/more classes
    - ⇨ Class- a class contains 1/more methods
    - ⇨
    - ⇨ <Suite>
    - ⇨ <test>
    - ⇨ <classes>
    - ⇨ <class />
    - ⇨ </classes>
    - ⇨ </test>
    - ⇨ </suite>

21. How to run failed test cases
    - ⇨ Once the execution is done, automatically test-output folder will be created in the project folder. In the test-output folder >>

testng-failed.xml file will be created. Right click on testng-failed.xml>> Run as TestNg. This will now only re-run the fail testcase(s)).

22. How to ignore packages and classes
23. ITestListener
    ⇨ Listeners are activated either before the test or after the test case. It is an interface that modifies the TestNG behavior. For example, when you are running a test case either through selenium or appium and suddenly a test case fails. We need a screenshot of the test case that has been failed, to achieve such scenario, TestNG provides a mechanism, i.e., Listeners. When the test case failure occurs, then it is redirected to the new block written for the screenshot.

24. Verify vs assert



Use of Listener for multiple classes.

If project has multiple classes adding Listeners to each one of them could be cumbersome and error prone.

In such cases, we can create a testng.xml and add listeners tag in XML.

testng.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Suite">

  <listeners>
  <listener class-name="Listener_Demo.ListenerTest"/>
  </listeners>

  <test name="Test">
     <classes>
       <class name="Listener_Demo.TestCases"/>
     </classes>
   </test> <!-- Test -->
</suite> <!-- Suite -->
```

This listener is implemented throughout the test suite irrespective of the number of classes you have. When you run this XML file, listeners will work on all classes mentioned. You can also declare any number of listener class.

⇨ Listeners are required to generate logs or customize TestNG reports in Selenium Webdriver.

25. TestNG listeners

⇨ Have capacity to listen to a specific incident.

- ⇨ TestNG is configured with listeners which can change default behavior of testNG.
- ⇨ TestNG listeners are used for logging purpose and creating reports.

26. TestNG vs Junit

- ⇨ TestNG →
  - → TestNG support to run parallel test cases
  - →supports adv annotations
  - →grouping tests together is possible in testNG
  - →dependancy tests r present in testNG
- ⇨ Junit →
  - → Junit does not support to run parallel test cases
  - ➔ Does not support adv annotations
  - ➔ Grouping tests together is not possible in JUNIT
  - ➔ Dependency tests r missing in Junit

Cucumber:

https://cucumber.io/docs/cucumber/

1. What is cucumber, adv, disadv, features

⇨ Cucumber is an **open-source software testing tool written in Ruby**. Cucumber enables you to write test cases that anyone can easily understand regardless of their technical knowledge.

⇨ Cucumber → Cucumber is a software tool used by the testers to develop test cases for the testing the behavior of the software.

⇨ In the Cucumber testing, the test cases are written in a simple English text, which anybody can understand without any technical knowledge. This simple English text is called the Gherkin language.

⇨ https://cucumber.io/docs/guides/parallel-execution/

⇨ Adv It is helpful to involve business stakeholders who can't easily read the code

⇨ Cucumber Testing enhances the end-user experience

⇨ Style of writing tests allow for easier reuse of code in the tests

⇨ Allows quick and easy setup and execution

2. Use of scenarioOutline and eg

⇨ The Scenario Outline keyword can **be used to run the same Scenario multiple times, with different combinations of values**.

⇨ Feature: Login into Application
⇨
⇨ Scenario Outline: Positive test validating login
⇨ Given Initialize browser with chrome
⇨ And Navigate to "https://www.flipkart.com/" site
⇨ And Click on login button in index page
⇨ When user enters <username> and <password>
⇨ Then verify user is successfully logged in
⇨ And close browser
⇨
⇨
⇨ Examples:
⇨ |username |password |

- ⇨ |username@example.com |password123 |
- ⇨ |test@abc.com |testpassword|
- ⇨

3. What is dataTable

- ⇨ Data tables are used when we need to test numerous input parameters of a web application. For example, the registration form of the new user involves several parameters to test, so for this, we can use the data table.
- ⇨ https://www.javatpoint.com/data-table-in-cucumber

4. Annotations/Hooks in cucumber

- ⇨ Cucumber supports only two hooks - @Before and @After
- ⇨ @Before → Before hooks run before the first step of each scenario.
- ⇨ @After → After hooks run after the last step of each scenario, even when the step result is failed, undefined, pending, or skipped.

5. Keywords(feature, and, when, then, given, but, scenario, back ground) and use in cucumber

- ⇨ Feature → The purpose of the Feature keyword is to provide a high-level description of a software feature, and to group related scenarios.

→The first primary keyword in a Gherkin document must always be Feature, followed by a : and a short text that describes the feature.

- ⇨ Scenario→One feature can have multiple scenarios, and each scenario consists of one or more steps.
- ⇨ Given→**Given** is a precondition. The use of Given keyword is to put the system in a familiar state before the user starts interacting with the system
- ⇨ Then→**Then** is testable outcome.

- ⇨ When→**When** is a user action.
- ⇨ And→Between any two statements, it gives the logical AND condition. AND can be combined with the GIVEN, WHEN, and THEN statements. *Example:* When I enter my account number AND CVV.
- ⇨ But→It denotes a logical OR relationship between two propositions. OR can be combined with the GIVEN, WHEN, and THEN statements. *Example:* Then I should be logged in BUT I must enter the OTP.
- ⇨ Background→Background keyword is used to group multiple given statements into a single group. The keyword mostly used when the same set of given statements are repeated in each scenario of the feature file.

6. What is gherkin lang

- ⇨ Gherkin is a readable business language that allows you to define business activity without getting bogged down in implementation specifics

7. What is stepDefination in cucumber, use of glue property

- ⇨ A step definition is the actual code implementation of the feature mentioned in the feature file.
- ⇨ The glue is a part of Cucumber options that describes the **location and path** of the step definition file

8. Files needed in cucumber framework and their use

- ⇨ Feature file → It has plain text descriptions of single or numerous test situations
- ⇨ stepDefinition → A step definition is the actual code implementation of the feature mentioned in the feature file

- ⇨ TestRunner → It connects feature file and stepDefinition file. It allows the user to run 1/more feature files at same time.

9. What is dry run

- ⇨ Cucumber dry run is **used for compilation of the Step Definition and Feature files and to verify** the compilation errors
- ⇨ The value of dry run can be either true or false. The default value of dry run is false and it is a part of the Test Runner Class file

10. How BDD framework works, aim of BDD framework, BDD VS TDD
11. Why to use cucumber with selenium
12. Jbehave vs cucumber
13. s/w require to run cucumber web test case
14. TestRunner class in cucumber

1. What is deferred state?
2. Test cases around time cases
3. Test cases around ecommerce websites from cart perspective.
4. Post method using get method

- I was just part of automating buttons, dropdowns and handling multiple elements

- What is Jenkins and why you used in your previous company:

  1. Jenkins is CI integration tool i.e. continuous integration tool. So in order to execute all my automation test cases I have used Jenkins.
  2. Basically I will be developing my test cases at local and then I will be pulling my test cases in Jenkins i.e. pulling my maven folder basically and then deploy build.

- What is GIT and how you used in your prev company
- If you have given web based application and stand alone application. What will you choose for performance testing?

  Ans: We cant do performance testing on stand alone applications.

- What is collection? And what are classes? And why there is need to use collection if there is Array concept?
- What is other name of page object model
- Can we use non-static methods in method overriding?

  Ans. Yes. **we cannot override static methods** because method overriding is based on dynamic binding at runtime and the static methods are bonded using static binding at compile time.

- In how many ways you can handle dropdowns?

  Ans:

  - one way is using select class, then by using.

- ByVisibleText/ByValue/ByIndex/if there is separate locator present for that web element the will use customize xpath.

- What is traceability matrix? Have you prepared document?
- What is conditional signoff?
- What is typecasting?upcasting and downcasting in terms of classs
- Error seeding
- Defect masking
- What is software testing?

  Ans. S/w testing is part of software development life process.

  It is an activity to detect and identify defects in software. The objective of testing is to release quality product to client

- What is s/w quality?

  Ans. Bug free, delivered on time, within budget, meets requirements and/or expectations, maintainable.

- Error, Bug and Failure

  Ans.

  Bug →which means that software or application is not working as per the requirement

  Error → The Problem in code leads to errors, which means that a mistake can occur due to the developer's coding error as the developer misunderstood the requirement

  https://www.javatpoint.com/bug-vs-defect-vs-error-vs-fault-vs-failure

❖ Why s/w has bugs?

- Mis-communication/no-communication
- s/w complexity
- prog errors
- frequent change in requirement
- lack of skilled testers

❖ Waterfall model – In this everything is documented and output of prev stage is input to next step. In waterfall, development of one phase starts only when the previous phase is complete… So if we change anything in requirement then we need to change everything from documentation.

Adv- quality of product will be good

- Since requirement changes are not allowed, chances of finding bugs will be less.
- Initial investment is less since the testers are hired at later stages.
- Preferred for small projects where requirements are freezed

Disadv of waterfall model

- Req changes are not allowed [if we make changes everytime then we will have to change documentation, design and implementation]
- If there is defect it is continued in later phases
- Testing will start after coding [here we have only one phase of testing, if we found any defect then we need to make changes in coding then in design and so on]

https://www.edureka.co/blog/software-testing-tutorial/

https://www.javatpoint.com/software-testing-bug-life-cycle

- What is UAT, System testing, Integration testing, and unit testing

  Unit testing- testing single module =>tested by developers.

  [Unit testing is a type of testing in which individual units or functions are tested. Its primary purpose is to test each unit or function. A unit is the smallest testable part of an application.]

  Integration module – after that they will integrate module (combine single modules and integrate one major module) and test that module

Unit testing and integration testing is done by developers. They test modules and sub-modules. Cz they know what r requirements and how it should work. And for Unit and integration testing programming/coding knowledge is required =>White box testing

System testing- tested by tester (in this verify if s/w is working as per client requirement or not.. comes under black box testing)

UAT – done by testers along with customers

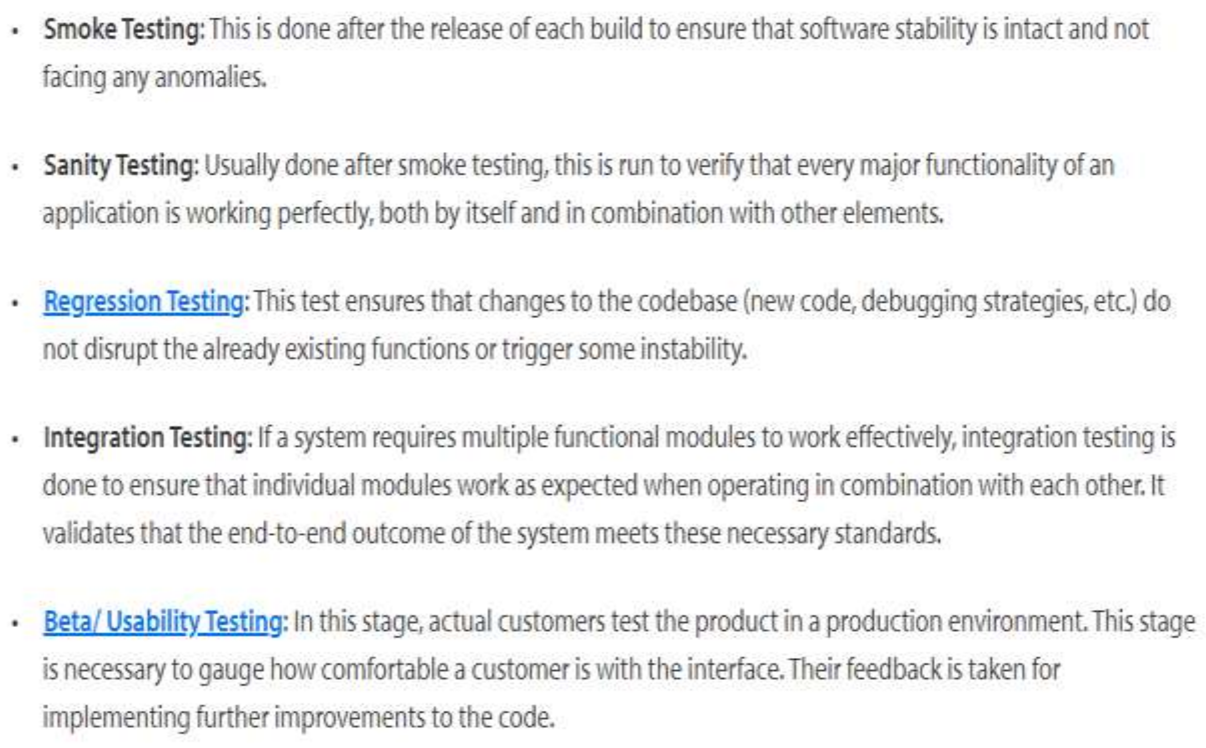
- What is verification and validation

  Validation is the **process of checking whether the specification captures the customer's needs**,

  while verification is the process of checking that the software meets the specification

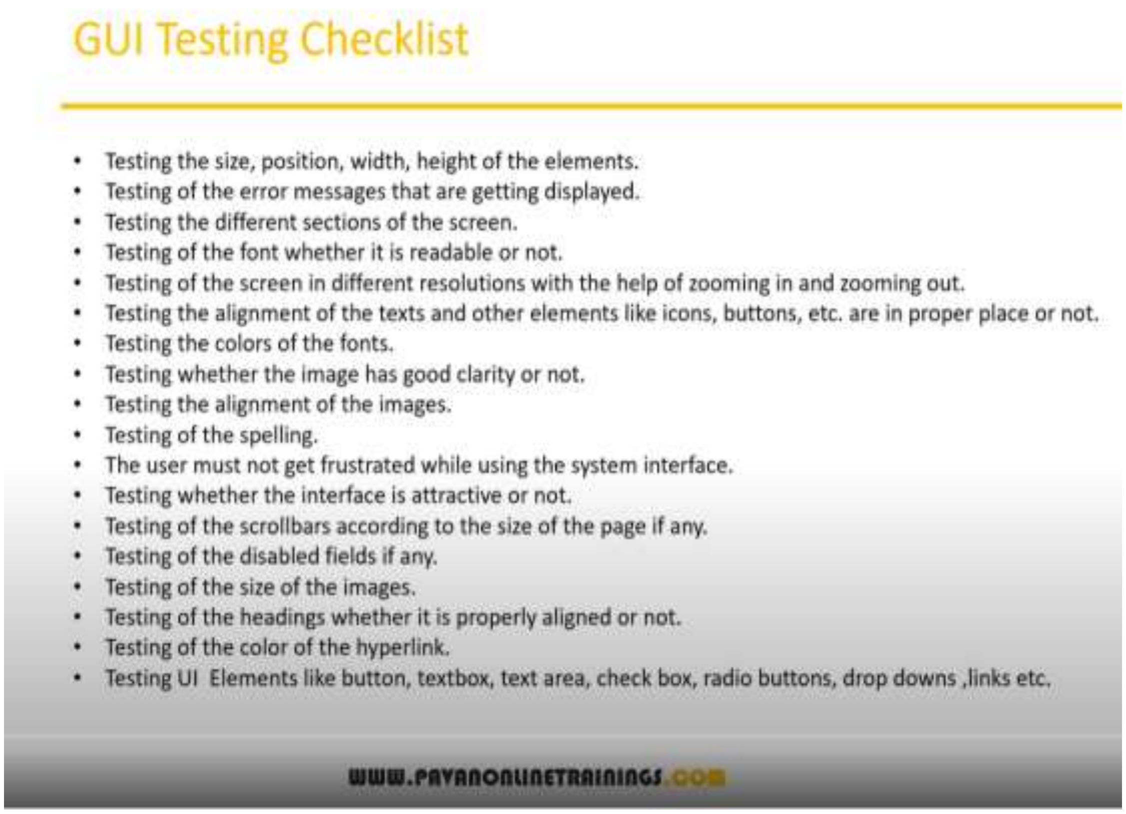
  - GUI Testing

- Testing user interface of application
- GUI includes elements like menu, check box, buttons, colors, fonts, size, icons, content and images



- **Smoke Testing:** This is done after the release of each build to ensure that software stability is intact and not facing any anomalies.
- **Sanity Testing:** Usually done after smoke testing, this is run to verify that every major functionality of an application is working perfectly, both by itself and in combination with other elements.
- **Regression Testing:** This test ensures that changes to the codebase (new code, debugging strategies, etc.) do not disrupt the already existing functions or trigger some instability.
- **Integration Testing:** If a system requires multiple functional modules to work effectively, integration testing is done to ensure that individual modules work as expected when operating in combination with each other. It validates that the end-to-end outcome of the system meets these necessary standards.
- **Beta/ Usability Testing:** In this stage, actual customers test the product in a production environment. This stage is necessary to gauge how comfortable a customer is with the interface. Their feedback is taken for implementing further improvements to the code.

-
- Alpha Testing is a type of software testing performed to identify bugs before releasing the product to real users or to the public. Alpha Testing is one of the user acceptance testing.
- Beta Testing is performed by real users of the software application in a real environment. Beta testing is one of the type of User Acceptance Testing.
-



GUI Testing Checklist

- Testing the size, position, width, height of the elements.
- Testing of the error messages that are getting displayed.
- Testing the different sections of the screen.
- Testing of the font whether it is readable or not.
- Testing of the screen in different resolutions with the help of zooming in and zooming out.
- Testing the alignment of the texts and other elements like icons, buttons, etc. are in proper place or not.
- Testing the colors of the fonts.
- Testing whether the image has good clarity or not.
- Testing the alignment of the images.
- Testing of the spelling.
- The user must not get frustrated while using the system interface.
- Testing whether the interface is attractive or not.
- Testing of the scrollbars according to the size of the page if any.
- Testing of the disabled fields if any.
- Testing of the size of the images.
- Testing of the headings whether it is properly aligned or not.
- Testing of the color of the hyperlink.
- Testing UI Elements like button, textbox, text area, check box, radio buttons, drop downs ,links etc.

WWW.PAVANONLINETRAININGS.COM

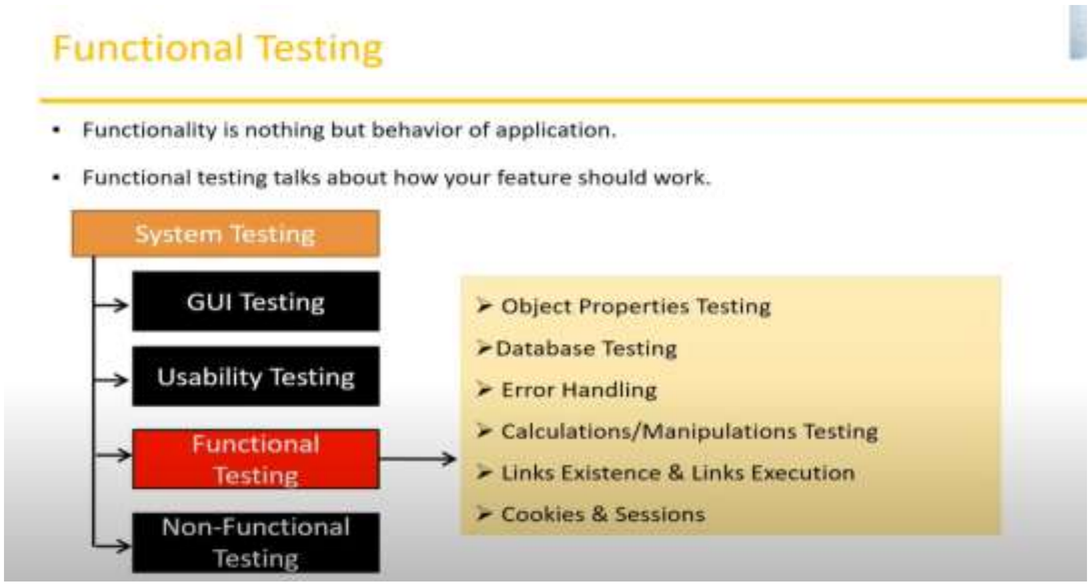
-

Usability Testing

- Check how easily end user are able to understand and operate the application. (i.e. validate if application is user friendly)

Functional Testing

-

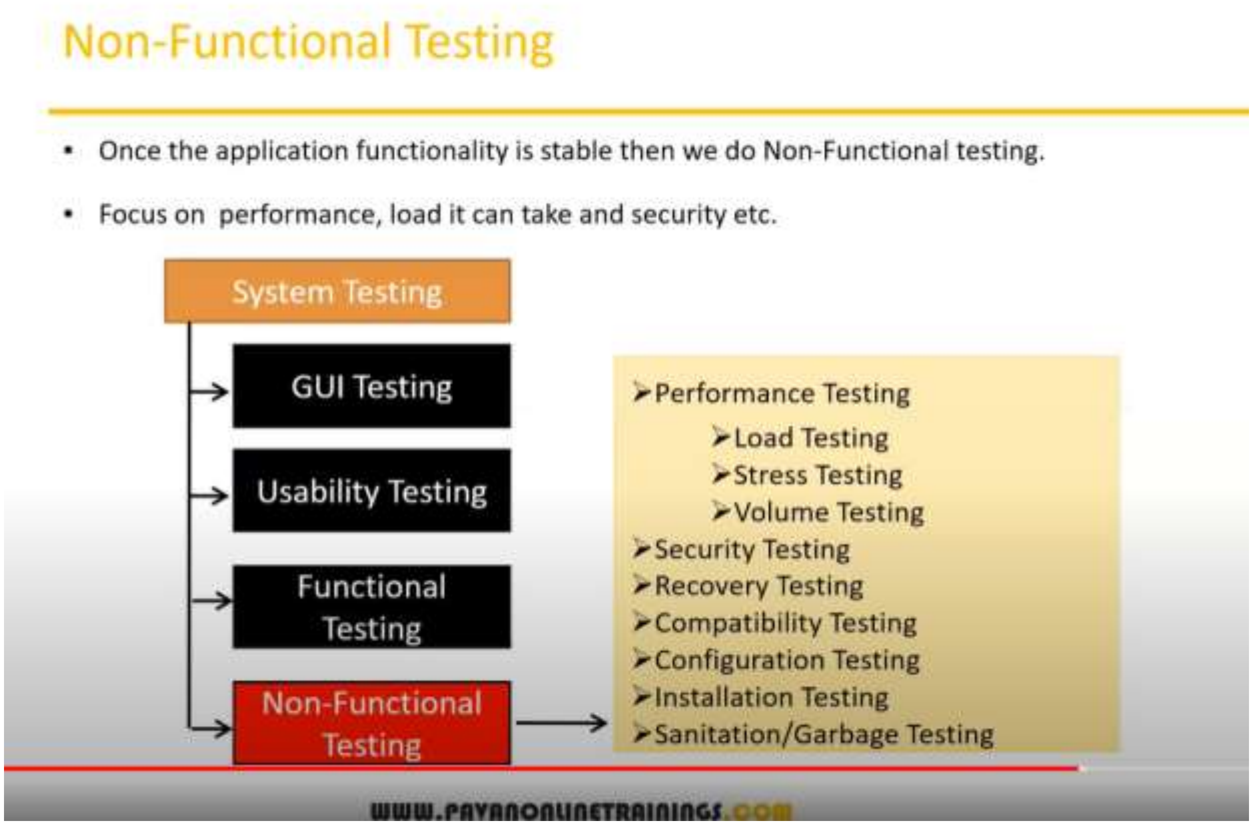


Functional Testing

- Functionality is nothing but behavior of application.
- Functional testing talks about how your feature should work.

- Object Properties Testing
- Database Testing
- Error Handling
- Calculations/Manipulations Testing
- Links Existence & Links Execution
- Cookies & Sessions

Non-functional -



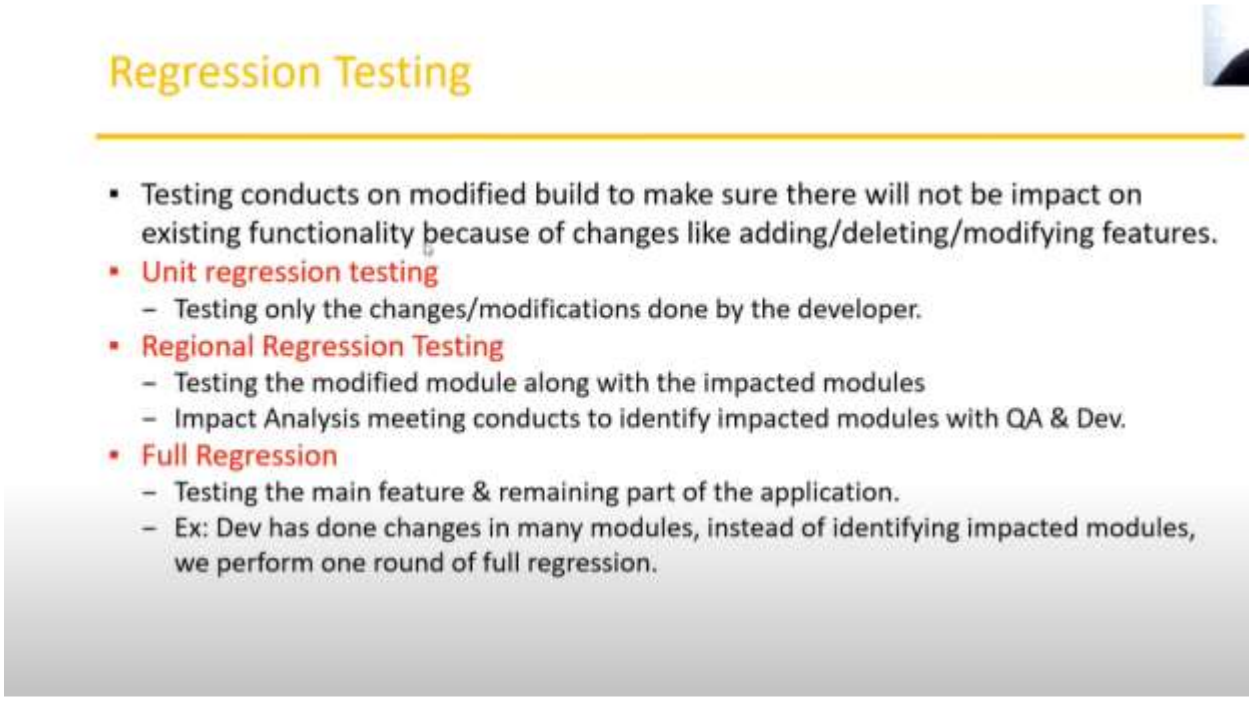
Non-Functional Testing

- Once the application functionality is stable then we do Non-Functional testing.
- Focus on performance, load it can take and security etc.

- Performance Testing
  - Load Testing
  - Stress Testing
  - Volume Testing
- Security Testing
- Recovery Testing
- Compatibility Testing
- Configuration Testing
- Installation Testing
- Sanitation/Garbage Testing

WWW.PAVANONLINETRAININGS.COM

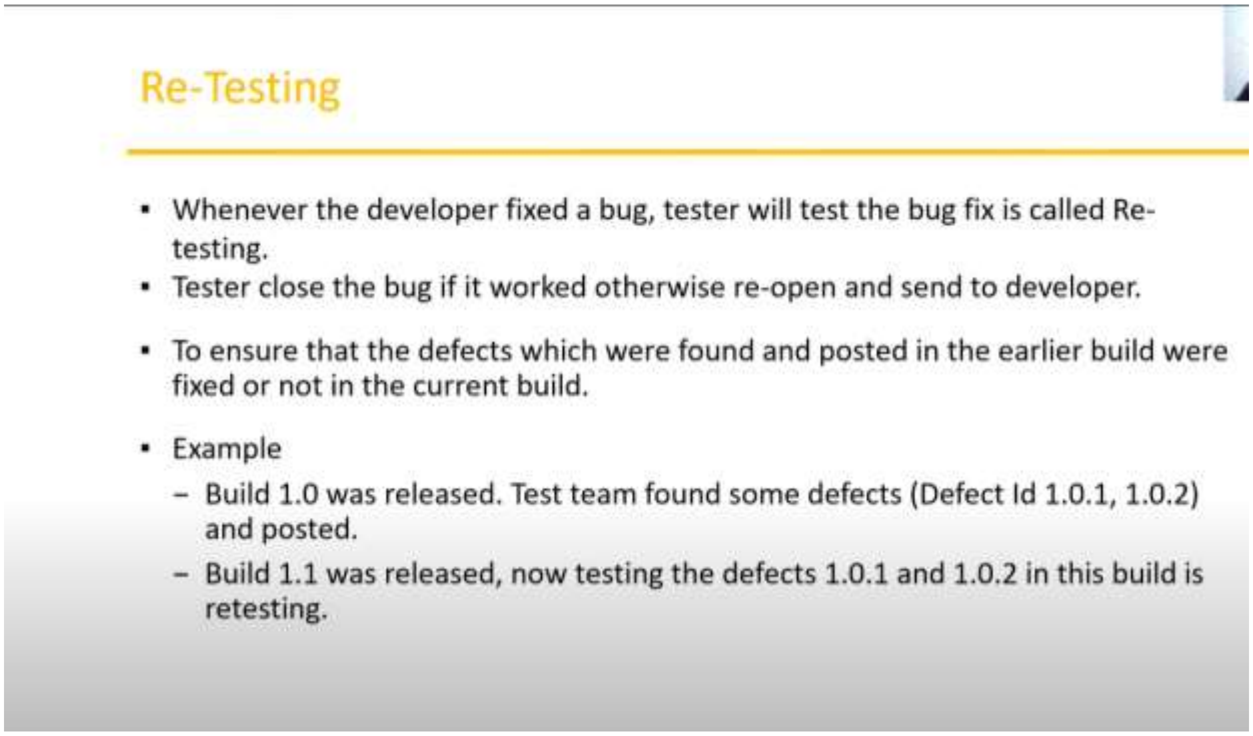
Regression Testing-



Regression Testing

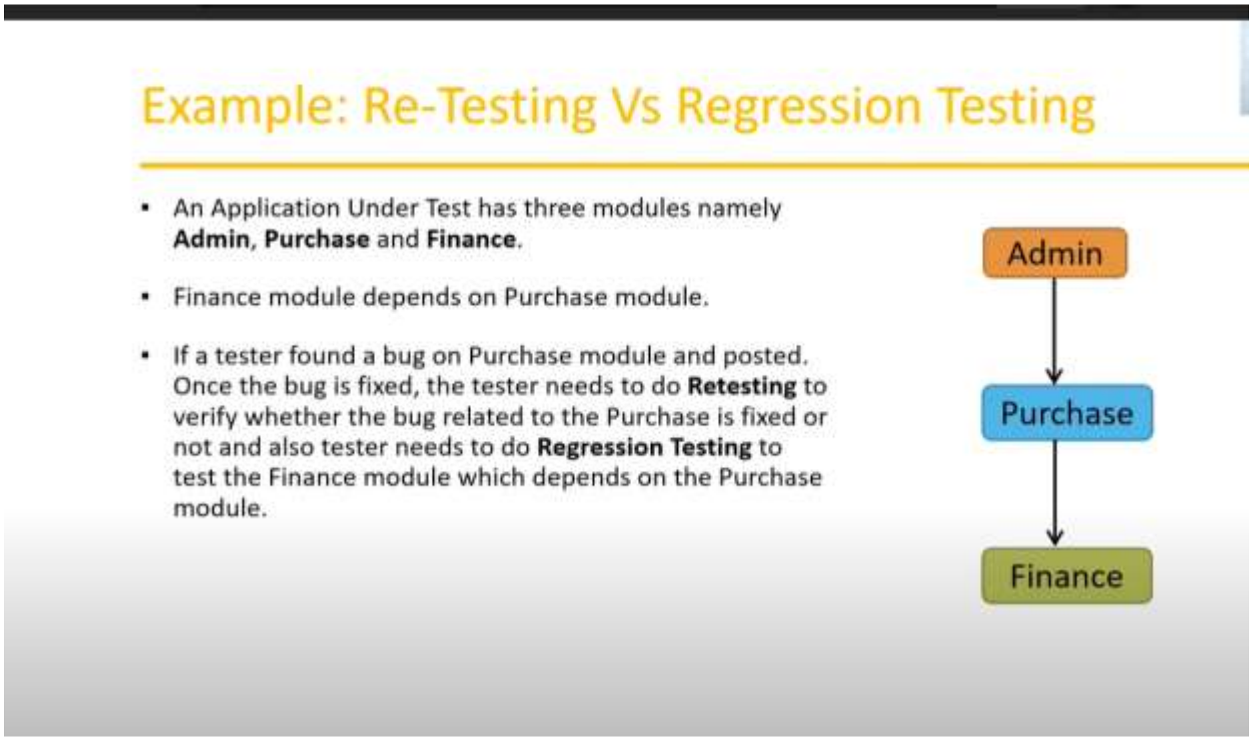
- Testing conducts on modified build to make sure there will not be impact on existing functionality because of changes like adding/deleting/modifying features.
- Unit regression testing
  - Testing only the changes/modifications done by the developer.
- Regional Regression Testing
  - Testing the modified module along with the impacted modules
  - Impact Analysis meeting conducts to identify impacted modules with QA & Dev.
- Full Regression
  - Testing the main feature & remaining part of the application.
  - Ex: Dev has done changes in many modules, instead of identifying impacted modules, we perform one round of full regression.

Re-Testing



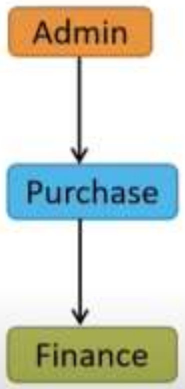
Re-Testing

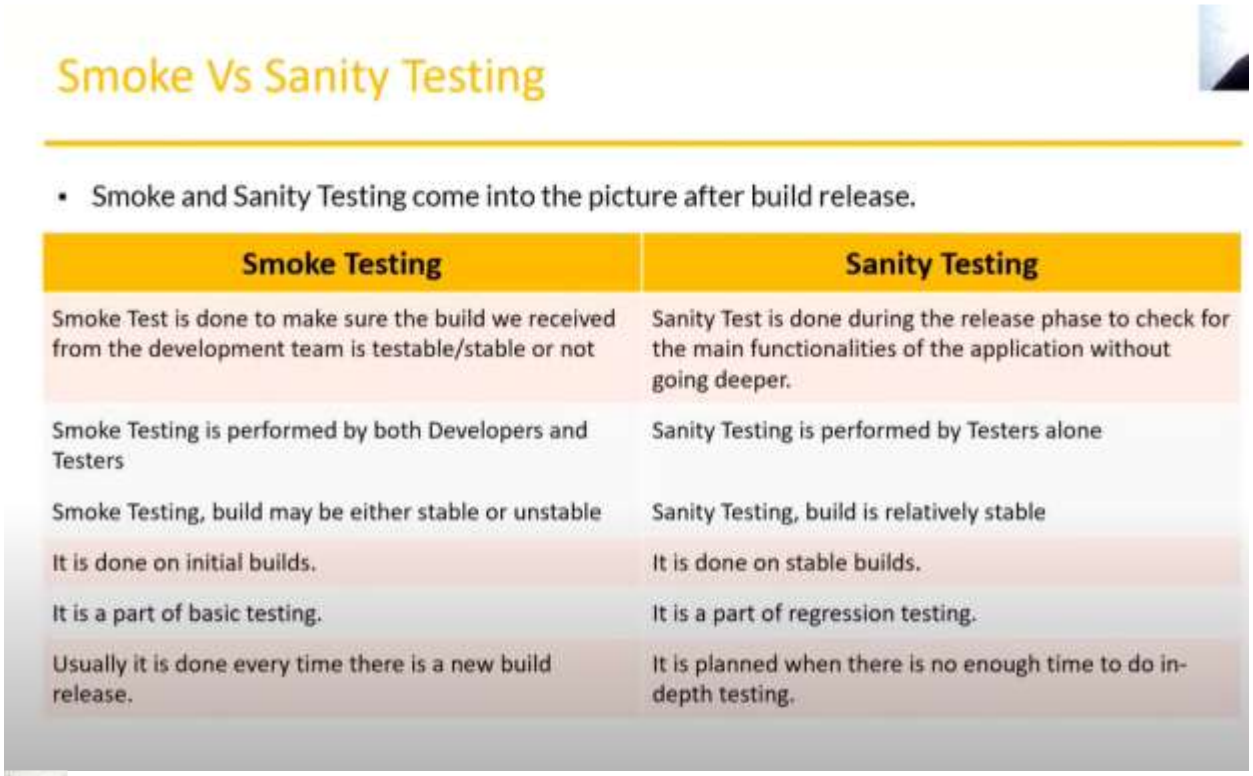
- Whenever the developer fixed a bug, tester will test the bug fix is called Re-testing.
- Tester close the bug if it worked otherwise re-open and send to developer.
- To ensure that the defects which were found and posted in the earlier build were fixed or not in the current build.
- Example
  - Build 1.0 was released. Test team found some defects (Defect Id 1.0.1, 1.0.2) and posted.
  - Build 1.1 was released, now testing the defects 1.0.1 and 1.0.2 in this build is retesting.



Example: Re-Testing Vs Regression Testing

- An Application Under Test has three modules namely **Admin**, **Purchase** and **Finance**.
- Finance module depends on Purchase module.
- If a tester found a bug on Purchase module and posted. Once the bug is fixed, the tester needs to do **Retesting** to verify whether the bug related to the Purchase is fixed or not and also tester needs to do **Regression Testing** to test the Finance module which depends on the Purchase module.



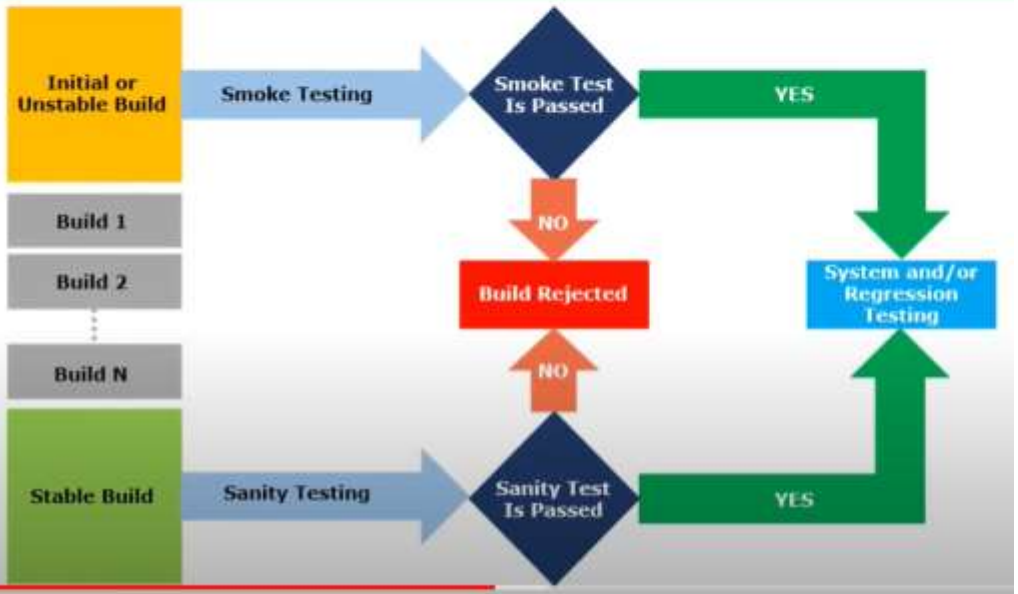


Smoke Vs Sanity Testing

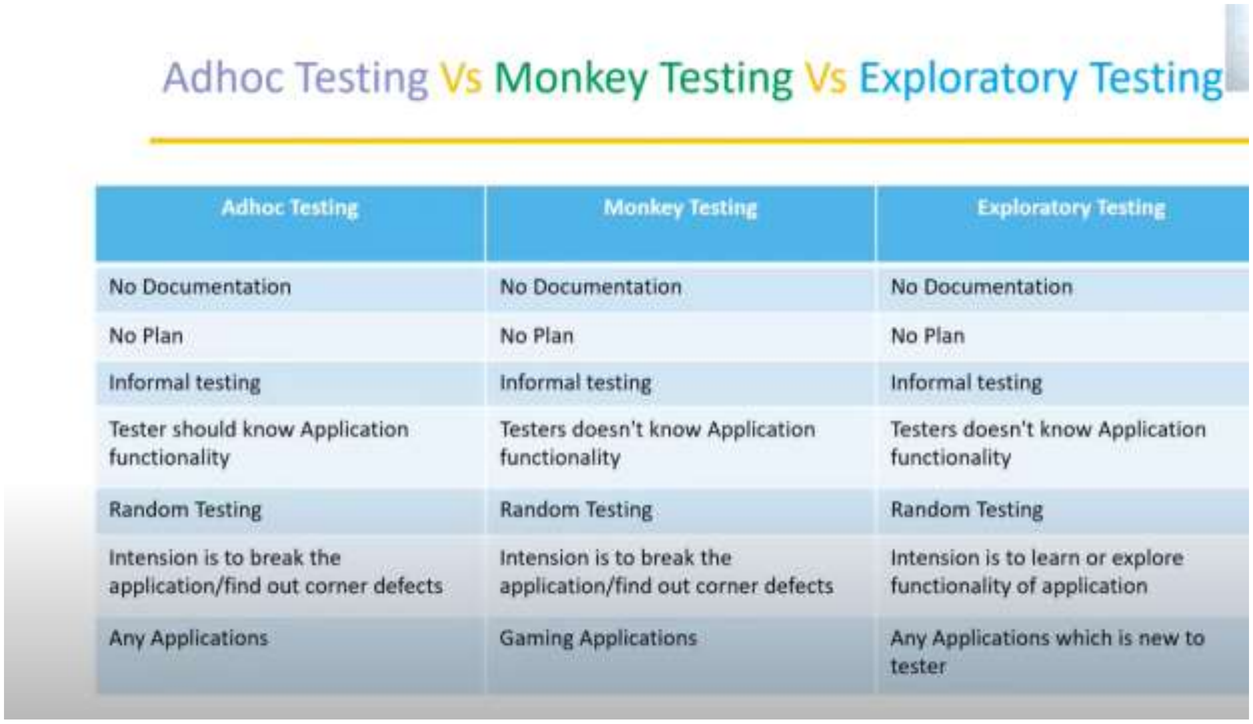
- Smoke and Sanity Testing come into the picture after build release.

Smoke Testing	Sanity Testing
Smoke Test is done to make sure the build we received from the development team is testable/stable or not	Sanity Test is done during the release phase to check for the main functionalities of the application without going deeper.
Smoke Testing is performed by both Developers and Testers	Sanity Testing is performed by Testers alone
Smoke Testing, build may be either stable or unstable	Sanity Testing, build is relatively stable
It is done on initial builds.	It is done on stable builds.
It is a part of basic testing.	It is a part of regression testing.
Usually it is done every time there is a new build release.	It is planned when there is no enough time to do in-depth testing.

Smoke Testing Vs Sanity Testing



WWW.PAVANONLINETRAININGS.COM


Adhoc Testing Vs Monkey Testing Vs Exploratory Testing

Adhoc Testing	Monkey Testing	Exploratory Testing
No Documentation	No Documentation	No Documentation
No Plan	No Plan	No Plan
Informal testing	Informal testing	Informal testing
Tester should know Application functionality	Testers doesn't know Application functionality	Testers doesn't know Application functionality
Random Testing	Random Testing	Random Testing
Intension is to break the application/find out corner defects	Intension is to break the application/find out corner defects	Intension is to learn or explore functionality of application
Any Applications	Gaming Applications	Any Applications which is new to tester

STLC - Software testing life cycle

1. Requirement analysis
2. Test planing
3. Test Design
4. Test execution
5. Defect error reporting/bug reporting/tracking
6. Test closure

Log4j – refer links.docx

Logging ➔ It is process of getting/fetching success, warning, errors/exceptions messages from server while it is running.

Log 4j ➔is a logging tool.

Log4j components ➔ 1. Logger 2. Appender 3. layout

1. **Logger**

2. **Appender** – Is used to specify where to store log messages

   Types are –

   a. ConsoleAppender -print log message on console

   b. FileAppender – store log message in .log file

   c. JdbcAppender – stores log messages inside DB table

   d. FTP/TelnetAppender – send data from one server to another

   e. SMTP – send log message using email

3. **Layout** – it indicates message format that should be printed on appender.

1. Simple layout – print msg as it is given by application

2. HTML layout – print msg as HTML file

3. XML layout – print msg as xml output

4. Pattern layout – print msg as given pattern by programmer

❖ What are advantages of log4j –

⇨ Log4j is open source

- ⇨ We can easily save log information into files/DB/Mail/Console(console used for only dev env)
- ⇨ Can be categorize at different levels(trace, debug, info, warn, error and fatal)
- ⇨ Can define format of output logs
- ⇨ Can save log with timestamp

  - ❖ What is Logger, appenders and layout
  - ❖ Levels of logging/ Priority methods–

    https://www.javatpoint.com/log4j-logging-levels

  - ⇨ Trace –
  - ⇨ Debug- is used to print Final/success msg. e.g. emp created with ID successfully.
  - ⇨ Info- current status in execution. E.g. try block execution completed
  - ⇨ Warn- any warnings in application e.g. file object is created but never closed, variable created but not used.
  - ⇨ Error – general/regular exceptions. NullPointerException, ArrayOutOfBoundException
  - ⇨ Fatal-
  - ⇨ Order --> Trace < debug <info < warn < error < fatal