1. **what is the use of framework in automation testing**

So, what exactly is an **automation framework**. In technical terms, an **automation framework** is a set of guidelines, which comprises of coding standards, structure of app code and app data folders, test results storage, accessing external resources and shared object repositories etc.

**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

1. **what are things u consider/keep in framework**

**http://www.evoketechnologies.com/blog/test-automation-framework-design/**

Below are some of the key parameters that a software tester needs to keep in mind, while developing a test automation framework.

* Handle scripts and data separately
* Create libraries
* Follow coding standards
* Offer high extensibility
* Less maintenance
* Script/Framework version control

1. **different types of frameworks we have in selenium/qtp**

Junit

TestNG

1. **what is data driven frame work and when do we use it?**

**Datadriven framework**- In this Framework , while Test case logic resides in Test Scripts, the Test Data is separated and kept outside the Test Scripts.Test Data is read from the external files (Excel File) and are loaded into the variables inside the Test Script. Variables are used both for Input values and for Verification values.

1. **what is keyword frame work and when do we use it?**

Keyword Driven framework- The Keyword-Driven or Table-Driven framework requires the development of data tables and keywords, independent of the test automation tool used to execute them . Tests can be designed with or without the Application. In a keyword-driven test, the functionality of the application-under-test is documented in a table as well as in step-by-step instructions for each test.

1. **what is hybrid framework and when do we use it**

**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.**what is POM and modular framework**

**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.

1. **what is testng and junit?**

Junit is a unit testing framework introduced by Apache. Junit is based on Java.

1. **writing unit test cases with testng**

Writing unit tests with TestNG is much like writing tests with JUnit: you create a class which contains methods annotated with the @Test annotation. Also, you can add methods that are run before or after each test or test case. One feature that is distinct to TestNG is the ability to group tests, making it possible to run only a certain group of tests instead of all.

All unit tests are placed in the same project they are testing, in a separate directory named test. This allows the tests to be compiled and run independently from the remaining code. It is good practice to have the same package structure for your tests and other code.

1. **what are important elements in testng.xml**

Answer: In a Selenium TestNG project, we use <testng.xml> file to configure the complete test suite into a single file. This file makes it easy to group all the test suites and their parameters in one file. It also gives the ability to pull out subsets of your tests or split several runtime configurations. Few of the tasks which we can group in the <testng.xml> file are as follows.

1- Can configure test suite comprising of multiple test cases to run from a single place.

2- Can include or exclude test methods test execution.

3- Can mark a group to include or exclude.

4- Can pass parameters in test cases.

5- Can add group dependencies.

6- Can configure parallel test execution.

7- Can add listeners.

1. **what are different annotations in testng and junit**

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.

**TestNG:**

@Test, @BeforeSuite, @AfterSuite, @BeforeTest, @AfterTest, @BeforeClass, @AfterClass, @BeforeMethod, @AfterMethod.

1. **what is group and suite and parallel execution in testng**

**https://www.tutorialspoint.com/testng/testng\_execution\_procedure.htm**

**Group test** is a new innovative feature in TestNG, which doesn’t exist in JUnit framework. It permits you to dispatch methods into proper portions and perform sophisticated groupings of test methods. Not only can you declare those methods that belong to groups, but you can also specify groups that contain other groups. Then, TestNG can be invoked and asked to include a certain set of groups (or regular expressions), while excluding another set. Group tests provide maximum flexibility in how you partition your tests and doesn't require you to recompile anything if you want to run two different sets of tests back to back.

Groups are specified in your testng.xml file using the <groups> tag. It can be found either under the <test> or <suite> tag. Groups specified in the <suite> tag apply to all the <test> tags underneath.

**A test suite** is a collection of test cases intended to test a behavior or a set of behaviors of software program. In TestNG, we cannot define a suite in testing source code, but it is represented by one XML file, as suite is the feature of execution. It also allows flexible configuration of the tests to be run. A suite can contain one or more tests and is defined by the <suite> tag.

<suite> is the root tag of your testng.xml. It describes a test suite, which in turn is made of several <test> sections.

1. **how to decide which test cases needs to be automated**

The right candidates for automation are:

* Regression test suite
* Smoke / Sanity test suite
* Build deployment
* Test data creation
* Automating behind the GUI like testing of APIs and methods

1. **what are the steps in automation testing or when do we automation testing?**

Ans: write manual test cases (if they are not there)

out of all manual test cases we will pick test cases that are of regression testing

(test cases that needs to be executed)

put all positive scenarios in regression

we dont put any UI related changes in regression testing

1. **what is the use of data provider annotation**

**16) Difference between junit and testng**

[**http://blog.varunin.com/2010/04/what-is-testng-and-whats-difference.html**](http://blog.varunin.com/2010/04/what-is-testng-and-whats-difference.html)

**17) how to generate reports using testng**

TestNG offers following two ways to produce a report.

**Listeners:** For a listener class, the class has to implement the org.testng./TestListener interface. TestNG notifies these classes at runtime when the test enters into any of the below states.

e.g. Test begins, finishes, skips, passes or fails.

**Reporters:** For a reporting class to implement, the class has to implement an org.testng/Reporter interface. When the whole suite run ends, these classes are called. When called, the object consisting the information of the whole test run is delivered to this class.

**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.