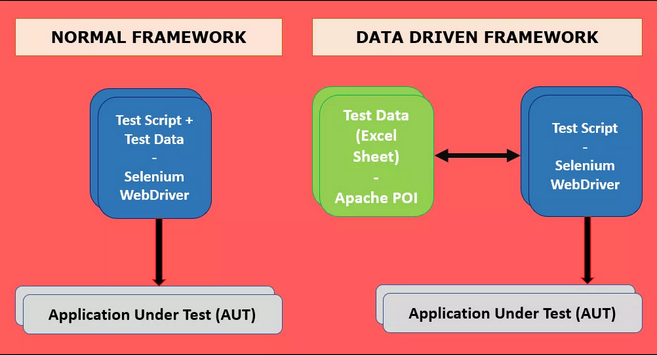
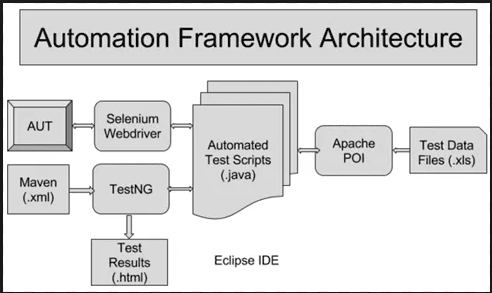
**Normal vs Datadriven Framework**





**What is the difference between Keyword-driven testing and Data-driven testing?**

In automated testing, to get larger test coverage in test automation, you can perform tests with different input of data. The basic working of Data-driven testing is that **you run your test with different set of input data** to ensure that with different various values, the application will perform as expected. In this method, you use data as inputs to your script actions. Each data set you have in the script, provides a test case. So the more data sets you provide, the more test cases you have.

Ex: Making appointment Function

On the other hand, **the basic working of Keyword driven testing is that you use a keyword to represent an action**. A sequence of keywords drives a script. So you can use the same set of keywords to build a variety of test scripts.

Ex: In this test case, all the keywords are specified and each keyword will have a description for ease of understanding.

**Any Framework Requirements**

* **Code Reuse**
* **Run on Different Environment**

**Components of Automation Framework**

* **Java**
* Maven
* Read Data from xlsx using POI
* Execute Tests (Selenium) and Validation (Assertions)
* Reports – XSLT, Extent
* Run Project with Ant/Maven/Eclipse
* Grid
* Jenkins integration – schedule runs
* Git

Core Data Drive Framework –

We will do Application testing after creating and integrating with Data Driven Framework

Note:

Steps for DD Framework

1. Create Maven Project
2. Create Packages in src/test/java
   1. ddf.base
   2. ddf.testcases
3. Create Dummy Tests
   1. TestA.java
   2. TestB.java
   3. TestC.java
4. Add dependencies and priorities between the tests e.g. TestA1, TestA2, TestA3

**(Note: Dependent tests should be in same java file)**

1. Create testing.xml for batch running
2. Create BaseTest class for writing reusable functions. This will have functions common to all Tests.java
   1. All Tests mentioned in point #3 should extend BaseTest.java

**Example functions to be added in BaseTest.java**

**Generic Functions**

**public** **void** openBrowser(String browserType)

**public** **void** navigate(String url)

**public** **void** click\_button(String button)

**public** **void** type(String locator, String text)

**public** WebElement getElement(String locator)

//Validations Functions

**public** **boolean** verifyTitle()

**public** **boolean** isElementPresent(String locator)

**public** **boolean** verifyText(String locator, String expectedText)

//Reporting Functions

**public** **void** reportPass(String msg)

**public** **void** reportCriticalFailure(String msg)

**public** **void** reportFailure(String msg)

**public** String takeScreenshot()

//Capturing screenshots

**public** **void** createScreenCaptureFromPath\_ifFail(String screenshotfilepath)

**public** **void** createScreenCaptureFromPath\_ifPass(String screenshotfilepath)

1. Initiate properties file
2. Create ‘util’ package in src/main/java
3. Generating different reports for each run depending on the time it was run