**What is TDD?**

TDD stands for test-driven development.it is an iterative development process.it starts with set of test written for a piece of functionality

Piece test are supposed to fail during first iteration, as there will be no application code.in the next phase the application code is written with an intention to pass the test case.

Once the application code is ready the test are run. If any failures the application code is refactored to make this test pass. Once the application code refactored the are run again. Once all test are passed we can ensure the features for which test were return have been developed

**BENEFITS OF TDD**

1. Unique test proves that the actually works refactoring allow to improve the design of the code

2. Test first will reduce the bugs

**DRAWBACKS**

1. Developer may think it is a waste of time re-writing the test when there is change in the requirement

**BDD**

BDD stands for BEHAVIOURAL DRIVEN DEVELOPMENT.BDD is an extension of TDD. like in TDD first we write the test and then add the application code the major difference between BDD and TDD are test are written in plain descriptive English language.

The test are explained as behaviour of application and are more user focus

**FEATURES OF BDD.**

1. Shifting from thinking of test to thinking of behaviour

2. It is easy to describe

Business stakeholders, developers interact each other to define the behaviour it acts as a bridge b/w business and technical people

**WHAT IS CUCUMBER?**

It is a testing framework, which supports BDD. IT helps us to define the behaviour in the plain English language knowns as "GHERKIN". Cucumber itself written in ruby language. It can be used to test the code, which is written in other language like java, python.

**GHERKIN**

**Feature:** Facebook login feature

**Scenario:** Login with valid username and password

**When:** user enter valid username and password

**And:** user click on login button

**Then: home** page to be displayed

To work with BDD cucumber we need to install

1. Cucumber eclipse plugin and

2. Natural plugin from eclipse market place has to be installed in eclipse

**List of jars to be downloaded**

1. Cucumber core

2. Cucumber HTML

3. Cucumber java

4. Cucumber jvm-deps

5. Testng

6. Gherkin

7. Selenium

**Cucumber Basics**

There are three main types

1. Feature file

2. Step definition

3. Runner class

**Feature file**

A feature file is an entry point to the cucumber test. It is where we define or we describe our test in simple English language called gherkin.

It is an essential part of cucumber as it serves the automation scripts as well as the live documents.

A feature file can contain a scenario or a list of scenarios based on feature what we are testing.

**HOW TO CREATE A FEATURE FILE.**

GO TO FILE-NEW-FILE-GIVE A FILE NAME WITH .FEATURE EXTENSION

***GHERKIN KEYWORDS***

**1. FEATURE**

Each gherkin file begins with feature keyword.

Feature defines the logical test functionality we will test in this feature file

**For example:**

If you are testing a login feature then the feature will be login.

**2. SCENARIO**

Each feature will contain some number of test-to-test the feature each test is called as scenario and it is defined by scenario keyword.

**3. GIVEN**

It defines pre-condition to the test

**4. WHEN**

It defines the test action that will be executed. Test action means the user inputs.

**5. THEN**

It defines the outcome of previous steps.

**6. AND**

It is used to define additional condition to the steps.

**7. BUT**

But keyword is used to add negative type of commands.

**8. BACKGROUND**

Given users navigates to Facebook website

When user validates the home page tittle

**CUCUMBER OPTIONS**

**FEATURES**

It helps cucumber to locate the feature file in the project folder structure

Feature= {“ WE HAVE TO PASS THE PATH HERE “ }

**GLUE**

Glue options helps cucumber to locate the path of step definition file

**FORMAT**

Format option is used to specify the different formatting options of the output

**Pretty**

It prints the gherkin source with additional colours and sack traces for errors

Format= {“pretty”}

**HTML**

This will generate the html report mentioned in the formatter it self

*SYNTAX:* Format= {“html: folder name”}

**JSON**

It is used to generate all the report in Jason format

SYNTAX: Format= {“json: folder name/cucumber.json”}

**DATA DRIVEN TESTING**

**1. Parametrizing without example keyword**

**Feature:** Facebook login feature

**Scenario:** Login with valid username=QWERT and password=QWERTY

*Given* user navigate to Facebook website

*When* user enter valid username and password

*And* user click on login button

*Then* home page to be displayed

**DATA DRIVEN TESTING USING EXAMPLE KEYWORD**

Scenario Outline: In order to verify login to Facebook

Given user navigates to Facebook website

Then user entered “<username>” username

And user entered “<password>” password

And user select the age category

| Age | location|

| Below 18 | INDIA|

| above18 | USA|

**Cucumber Cukes:**

Cucumber Tax

@sanity @prodnew

Feature login

@smoke

Scenario outline

Example

@cucumber options

**TAGS**

**OR TAG:**

*Syntax*: tags={“ @smoke, @sanity”}

**AND TAG:**

*Syntax:* tags= {“@smoke”, ”@sanity”}

**Cucumber Hoops**

Example

@before ()

Public void setup ( )

{

System.out.println (“launch the browser”);

}

@after

Public void teardown ( )

{

System.out.println (“driver. quit( )”);

}