**Behavior Driven Development**

Behavior Driven testing is an extension of TDD. Like in TDD in BDD also we write tests first and the add application code. The major difference that we get to see here are

* Tests are written in plain descriptive English type grammar
* Tests are explained as behavior of application and are more user focused
* Using examples to clarify requirements

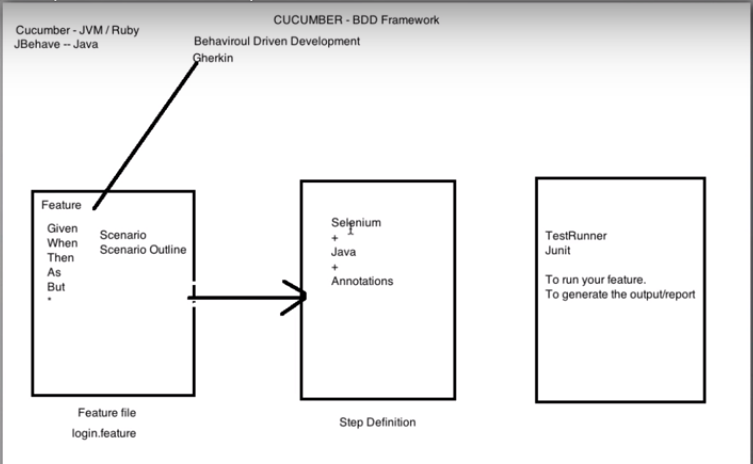
**Features of BDD**

1. Shifting from thinking in “tests” to thinking in “behavior”
2. Collaboration between Business stakeholders, Business Analysts, QA Team and developers
3. Ubiquitous language, it is easy to describe
4. Driven by Business Value
5. Extends Test Driven Development (TDD) by utilizing natural language that non technical stakeholders can understand

BDD frameworks such as Cucumber or JBehave are an enabler, acting a “bridge” between Business & Technical Language

### **What is Cucumber?**

* **Cucumber** is a testing framework which supports **Behavior Driven Development (BDD).**It lets us define application behavior in plain meaningful English text using a simple grammar defined by a language called **Gherkin**.
* Cucumber itself is written in **Ruby**, but it can be used to “test” code written in *Ruby* or other languages including but not limited to *Java*, *C#* and *Python.*



# **Install Cucumber Eclipse Plugin:**

1) Launch the *Eclipse IDE* and from Help menu, click “***Install New Software***”.

2) You will see a dialog window, click “***Add***” button.

3) Type name as you wish, let’s take “***Cucumber***” and type “***http://cucumber.github.com/cucumber-eclipse/update-site***” as location. Click ***OK***.

4) You come back to the previous window but this time you must see ***Cucumber Eclipse Plugin*** option in the available software list. Just ***Check*** the box and press “***Next***” button.

5) Click on ***Next and Finish***.

**Add maven dependencies:**

Cucumber-java

Cucumber-jvm

Cucumber-junit

Cucumber-jvm-deps

Cucumber- reporting

Gherkin

Junit

**Add tool from market place:**

Natural

## Cucumber Feature File

A **Feature File** is an entry point to the *Cucumber* tests. This is a file where you will describe your tests in Descriptive language (Like English). It is an essential part of Cucumber, as it serves as an automation test script as well as live documents. A feature file can contain a scenario or can contain many scenarios in a single feature file but it usually contains a list of scenarios.

## Gherkin

Gherkin is a structured language it follows some syntax let us first see a simple scenario described in gherkin.

**Feature:** Search feature for users This feature is very important because it will allow users to filter products

**Scenario:** When a user searches, without spelling mistake, for a product name present in inventory. All the products with similar name should be displayed

**Given** User is on the main page of www.myshopingsite.com  
**When** User searches for laptops  
**Then** search page should be updated with the lists of laptops

**Runner Class:**

Now that we have defined the test its time to run our test. But before we do that we have to add a class for running our tests. Cucumber uses Junit framework to run. As *Cucumber* uses *Junit* we need to have a **Test Runner class**. This class will use the *Junit annotation* **@RunWith(),**which tells *JUnit* what is the *test runner class*. It more like a starting point for Junit to start executing your tests.

@RunWith(Cucumber.**class**)

@CucumberOptions(

features = “path”, // to pass path of the feature files

glue = {“path”}, // to pass path of the step definition files

## Step Definition:

A Step Definition is a small piece of *code* with a *pattern* attached to it or in other words a Step Definition is a java method in a class with an annotation above it. An annotation followed by the pattern is used to link the *Step Definition* to all the matching *Steps*, and the *code* is what *Cucumber* will execute when it sees a *Gherkin Step*. *Cucumber* finds the *Step Definition* file with the help of Glue code in Cucumber Options.

<dependency>

<groupId>info.cukes</groupId>

<artifactId>cucumber-java</artifactId>

<version>1.2.5</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>info.cukes</groupId>

<artifactId>cucumber-jvm</artifactId>

<version>1.2.5</version>

<type>pom</type>

</dependency>

<dependency>

<groupId>info.cukes</groupId>

<artifactId>cucumber-junit</artifactId>

<version>1.2.5</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>info.cukes</groupId>

<artifactId>cucumber-jvm-deps</artifactId>

<version>1.0.5</version>

</dependency>

<dependency>

<groupId>net.masterthought</groupId>

<artifactId>cucumber-reporting</artifactId>

<version>1.0.0</version>

</dependency>

<dependency>

<groupId>info.cukes</groupId>

<artifactId>gherkin</artifactId>

<version>2.12.2</version>

</dependency>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>3.5.3</version>

</dependency>

<!-- https://mvnrepository.com/artifact/info.cukes/cucumber-picocontainer -->

<dependency>

<groupId>info.cukes</groupId>

<artifactId>cucumber-picocontainer</artifactId>

<version>1.2.5</version>

<scope>test</scope>

</dependency>