**Cucumber**

**BDD** in a nutshell Behaviour-Driven Development (BDD) is a set of practices that enables software teams to produce more valuable software with fewer bugs. BDD can lead **to reduced development and maintenance costs and shorter time to market.**

The two main practices of BDD are **Specification by Example** and **Test-Driven Development(TDD).**

(BDD- TDD+ Incremental Requirement Analysis)

**Specification by Example:**

Specification by Example (SbE) is a technique that enables product owners, business analysts, testers and programmers to eliminate common misunderstandings about business requirements. SbE uses examples in conversations to illustrate business rules and behaviour of the software to be built.

**Test-Driven Development:**

Test-Driven Development (TDD) in the context of BDD turns examples into human-readable executable specifications. **Programmers use these specifications as a guide to implement increments of new functionality.** The result is a lean codebase and a suite of automated regression tests that keeps maintenance costs low throughout the lifetime of the software.

**Cucumber** lets you keep specifications, automated tests and documentation in the same place - a single source of truth that never gets out of sync.

Other popular BDD f/ws: JBehave, Jasmine etc

Login – 10 testcases-

basic code – run 10 test cases – 6 of them , 4 test cases failed

may be +3 testcases based on above failure

modify code- run 10 test cases – 9 of them, 1 test case failed

modify code- 10 testcases passed

Tests are driving the devlopement – TestDrivenDevelopment(TDD)

**Cucumber flow:**

1. **Runner class** – where we configure path to feature files and step definitions

Runner uses JUnit framework.

1. **Feature file:**

file which has extension .feature

File where you describe the requirements as small scenarios in **simple english language(Gherkin language**- internally uses Ruby)

We use simple English keywords like **given, when, then** to explain the scenarios

based on **gherkin** language

A single feature file can have more than one scenario.

UserStory or **feature**

eg: Customer Registration

username, pwd, email, phone(mandatory)

age,DOB, address(optional)

Testcases or **scenarios**:

ValidScenario with required fields

ValidScenario with required fields and optional fields

InvalidScenario without required fields

InvalidScenario for validations- password character

……

**3.StepDefinition classes:**

**Every feature file will have corresponding step definition class**

**It’s a Java class which is connected with respective feature file**

**When you run/execute a feature file- it executes the logic inside stepdefinition**

**class**

**Setup:**

1. **Install cucumber eclipse plugin:**

Update Site- copy below link to eclipse- install new software- add-

<https://cucumber.io/cucumber-eclipse/update-site>

1. **Create new maven project and add required dependencies**

**Cucumber can be used or integrated with any automation tools:**

Web automation using WebDriver

API automation using RestAssured or HttpClient

Mobile Automation using Appium

**Cucumber Before annotation runs before every scenario.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**LoginScenario**

**enter username\enter pwd**

**click submit**

**execute testcases**

**test cases fail**

**7testcases will pass**

**8 testcases fail**

**12 , 3**

**15 testcases passed**

**tests are driving development**

**TDD- Test Driven Development**

**BDD- Behaviour driven development**

**feature file-**

**requirements which are split into scenarios**

**Feature – Gherkin**

**added cucumber eclipse plugin**

**added maven dependencies after creating maven project**

**we created RunCukesTest- config class**

**we create feature file which is connected stepdefintion(java class)**

**you can integrate cucumber with webdriver/restassured/httpclient/appium**

**It depends on what kind of testing we want to do- UIAutomation/APIAutomation/MobileAutomation**