**BDD (Behavior driven development)**

* BDD is all about collaboration between teams
* We will be usually having two types of teams:

1. Business teams
2. Technical teams

* BDD helps in building a common understanding about the behavior of the application among the teams.
* The BDD will help in generating the common documentation that will help to make the stakeholders and other teams to understand.
* This documentation will serve as the single source of truth for all the teams in the organization.
* Say if the stakeholder will have to require **any enhancements** in the application then the **user story** will be created using **agile** process.
* User story I nothing but the informal, general explanation of the software feature written as per the requirement of end user or stake holder.
* Once the user story is created

1. This will be discussed among the teams
2. Then conversation and discussion among the teams and decide how the system should behave
3. Examples on this will be created, discussed, agreed and approved.
4. These examples are been documented in a way that can be developed and tested with automation.
5. Then we finally go to the coding phase, where in the implementation will be done based on the documentation examples.

* There are **3 practices of BDD**:

1. Discovery
2. Formulation
3. Automation

|  |  |  |
| --- | --- | --- |
| Discovery | Formulation | Automation |
| User story will be discussed among the teams.  Outcome:  Agreed behavior of the system. | Examples are been created based on the agreed behavior.  Outcome:  Documentation of examples. | Coding as per the documented examples to implement the agreed behavior of the system.  Create automation tests taking one example at a time.  Outcome:  Implemented code and automation tests. |

* Note: these documented examples are also called living documentations.
* Which is outcome of teams shared understanding.
* Which gives out or guides development and automation.
* So in future if again any changes are required then teams can view this documentation for their needs.

**Cucumber**

* Cucumber is a tool that facilitates the BDD.
* There are multiple tools to carry on BDD.
* So if we are in organization we use cucumber to automate the documentation.
* Cucumber is a tool that understands the documentation and turns it into a automated tests.

Firstly let us know about the evolution:

TLD:

* Test last development
* First the code is written then the test cases are.
* Which were basically used very earlier

TDD:

* Test driven development
* Here first the test cases are written
* When run these cases fail
* Then write the code to pass the test case.
* Refactor tests to remove redundancies.

BDD:

* BDD is same as TDD.
* But it is an extension of TDD.
* Every requirement or the feature will be tested.
* The user stories or the requirements are written in shared languages (gherkin).
* To make the user story understand by both technical and non-technical teams the cucumber has introduced common language called Gherkin.

Example for gherkin language:

**Feature**: login functionality

**Scenario**:

**Given** user is on login screen.

**When** user enters the username and password

**And** clicks on submit button

**Then** user will be taken to the home page

* Note that here the bolded words are keywords
* This gherkins help in braking down the user story into atomic actions
* Keeps the format consistent – that is it should be understood by all members of organization and also stakeholder.
* Helps in automating the scenarios

**TOOLS FOR BDD:**

* **CUCUMBER**
* **JBehave**
* **Behat**

Sprint is the short time period where the team works to complete the set amount of work.

Before sprint:

* We have two phases that is discovery and formulation

During sprint:

* Develop and automation arrives.
* After which test execution and validation and release of the feature or user story.