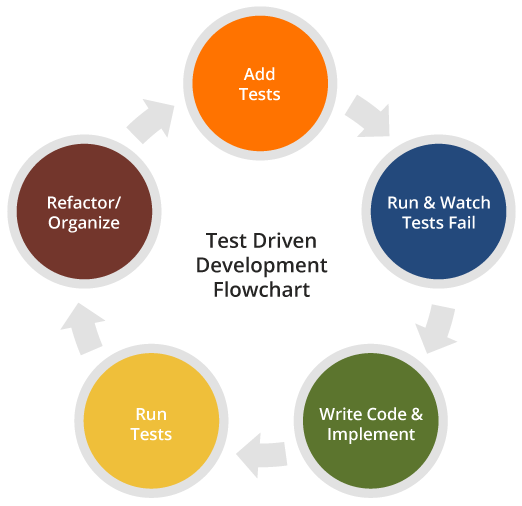
**Test Driven Development vs. Behavior Driven Development**

Chances are you have heard of test driven development (TDD) as well as behavior driven development (BDD) in regards to software development. Below are definitions of both these practices as well as the major differences between the two.

**Test Driven Development (TDD)**

TDD is a software development technique that involves writing automated test cases prior to writing functional pieces of the code. This is popular in agile methodologies as it drives delivering a shippable product at the end of a sprint. This process can be divided into multiple steps:

1. A developer, based on requirement documents, writes an automated test case.
2. The development team runs these automated test scripts against what is currently developed and the tests fail, as they should since none of the features have been implemented yet.
3. Development team functional code to ensure the automated test script gives them a green light.
4. The development team can then refactor and organize the code to produce a tested deliverable at the end of the sprint.



Test cases are mostly written in programming languages such as Java, Ruby, etc. and can be written using test automation tools such as Selenium, Watir, Windmill, etc. Since test scripts are written in programming languages, it is hard for a business analyst or test owner to verify the test scripts.

**Behavior Driven Development (BDD)**

BDD is a software development technique that defines the user behavior prior to writing test automation scripts or the functional pieces of code. Used in an agile sprint, this method ensures that a shippable product is generated at the end of a sprint. This involves:

1. Behavior of the user is defined by a product owner/business analyst/QA in simple English.
2. These are then converted to automated scripts to run against functional code.
3. The development team then starts writing the functional code to ensure the automated test script gives them a green light.
4. The development team can then refactor and organize the code to produce a tested deliverable at the end of the sprint.

BDD can be driven by multiple tools such as Cucumber, FitNesse, PowerTools, Docker, etc. The test scripts are written in plain English in Gherkin, Wiki frameworks, etc. Since the behavior is defined in English, it gives a common ground for ALL stakeholders involved in the project. This reduces the risk of developing code that wouldn’t stand up to the accepted behavior of the user.

BDD explains the behavior of an application for the end user while TDD focuses on how functionality is implemented. #Dev

**TDD vs. BDD**

1. BDD is in a more readable format by every stake holder since it is in English, unlike TDD test cases written in programming languages such as Ruby, Java etc.
2. BDD explains the behavior of an application for the end user while TDD focuses on how functionality is implemented. Changes on functionality can be accommodated with less impact in BDD as opposed to TDD.
3. BDD enables all the stakeholders to be on the same page with requirements which makes acceptance easy, as opposed to TDD.