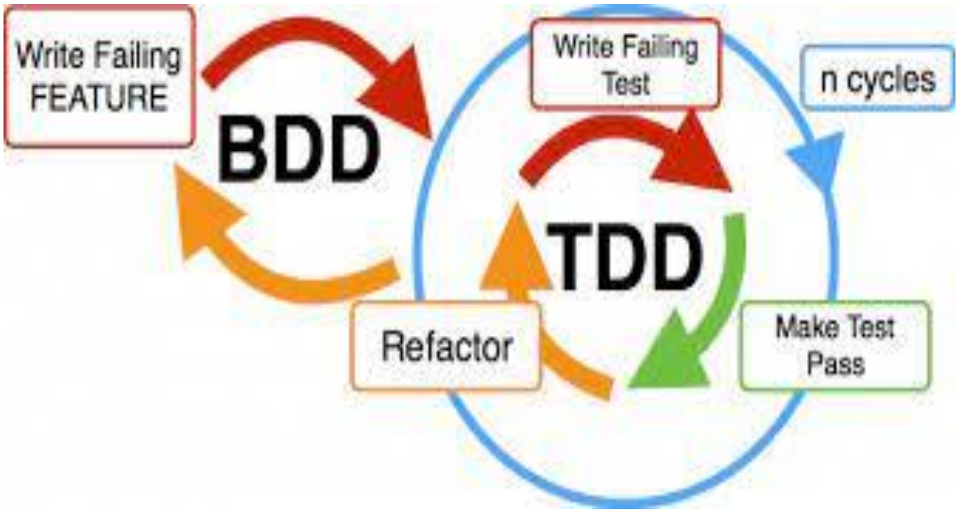
BDD : (Behavior Driven Development)

BDD is a software development process that originally emerged from Test Driven Development (TDD). BDD uses examples to illustrate the behavior of the system that are written in a readable and understandable language for everyone involved in the development.



How behavior-driven development works

Behavior-driven development revolves around conducting behavior-specific tests, or functional specifications that outline executable scenarios for the application. This includes:

- Applying the 5 Whys principle or the if-then scenario to generate user stories and clearly relate application features to a business purpose.
- Identifying a single outcome for every behavior.
- Translating each scenario into domain specific language (DSL) to ensure accurate communication.
- Gathering all behaviors into one set of documentation so it is accessible for all developers, testers and stakeholders.

Behavior-specific tests can be run when a project starts, while a product is in development and when a product is completed. At a minimum, BDD requires that the behavioral tests (similar to unit tests) are created before development starts. Before development begins, all of the behavioral tests will fail, but as development of the product progresses, the tests will begin to pass.

Advantages of BDD :

- Reduce waste: BDD is focused on discovering and developing features that brings high business value, so any feature that does not bring value will be ignored.

- Reduce costs: Because of the reduced waste, it means that the development process will be focused just on the features that matters, so any additional cost is reduced.

- Easier and safer changes: The requirements changes will be easier to implement because of the living documentation and safer because of the executable specification, which will act as automated acceptance and unit tests.

- Faster releases: The releases will become more frequent as the large number of automated tests will eliminate some of the manual testing done by the testers. The testers will have instead more time to focus on other more complicated tests and scenarios.

Disadvantages of BDD:

- BDD requires high communication and collaboration: BDD practices are very focused on discussions and collaboration between all parties involved. In case the business owners are unwilling or unable to engage in conversions and give feedback in time, then it will be hard to take advantage of BDD.

- BDD works best in an Agile or iterative context: One of the BDD principles assume that it is very hard to know all the requirements from the beginning and that we don’t have to define them all in the first phase of a project, but instead the knowledge of the stakeholders will evolve during the lifetime of a project. That is why it is considered that BDD is well suited to be used in conjunction with Agile or iterative methodologies.

- BDD does not work not well in an enclosed environment: In some cases, for example large projects developed by corporations, which delegates the work to multiple remote teams, will be harder for them to enable the high collaboration between teams.

- Poorly written tests can lead to higher test-maintenance costs: For some complex applications, it will require experience to design and write the automated acceptance tests. In case the application will acquire many poorly written tests, they will become hard to maintain.