Day3 Assignment 2 :-

Produce a comparative infographic of TDD, BDD, and FDD methodologies. Illustrate their unique approaches, benefits, and suitability for different software development contexts. Use visuals to enhance understanding.

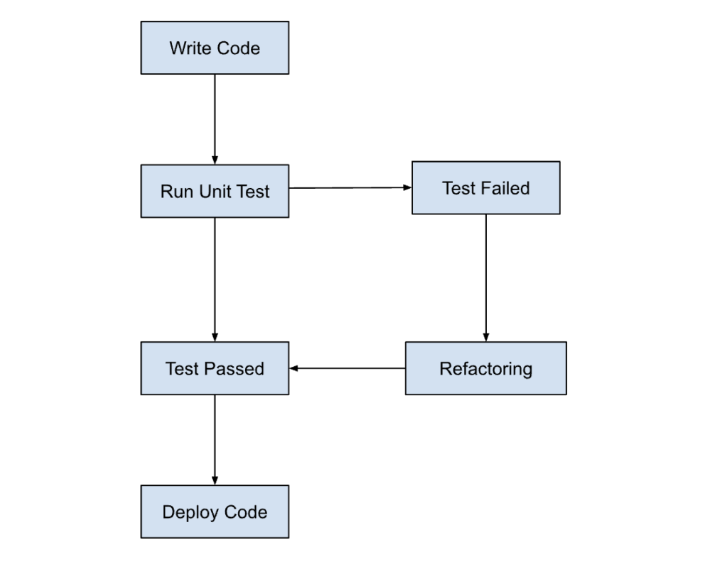
—---------------------------------------------------------------------------------------------------------

**Test-Driven Development [ TDD ]**

Test-Driven Development is a testing methodology or a programming practice implemented from a developer’s perspective. In this technique, a QA engineer starts designing and writing test cases for every small functionality of an application.

The main intention of this technique is to modify or write a fresh code only when the test fails. Hence it results in lesser duplication of test scripts. This technique is prevalent mainly in agile development ecosystems. In a TDD approach, automated test scripts are written before functional pieces of code. The TDD methodology involves the following steps:

1. A developer writes an automated test case based on the requirements specified in the documents.
2. These tests are executed, and in some cases, they fail as they are developed before the development of an actual feature.
3. The development team then re-factors the code for the test to pass successfully.
4. TDD can be done by a single developer while writing both tests and application code side by side to complete a feature.



Benefits of TDD

1. Reduces the amount of time required for rework
2. Explores bugs or errors very quickly
3. Faster feedback
4. Encourages the development of cleaner and better designs
5. Enhances the productivity of the programmer
6. Allows any team member to start working on the code without a specific team member. This encourages knowledge-sharing and collaboration.
7. It gives the programmer confidence to change an application’s large architecture quickly.

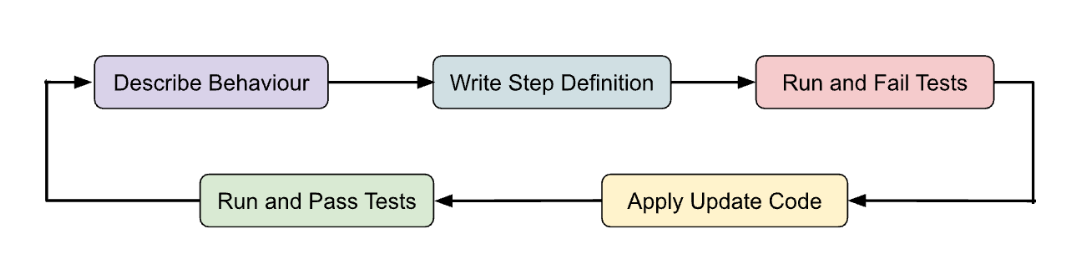
**Behavioral-Driven Development (BDD)**

Behavioral-Driven Development (BDD) is a testing approach derived from the Test-Driven Development (TDD) methodology. In BDD, tests are mainly based on systems behavior. This approach defines various ways to develop a feature based on its behavior. In most cases, the Given-When-Then approach is used for writing test cases. Let’s take an example for a better understanding of TDD vs BDD:

Given the user has entered valid login credentials

When a user clicks on the login button

Then display the successful validation message



Key benefits of BDD

1. Helps reach a wider audience through the usage of non-technical language
2. Focuses on how the system should behave from the customer’s and the developer’s perspective
3. BDD is a cost-effective technique
4. Reduces efforts needed to verify any post-deployment defects

How does BDD streamline SDLC :-

Debugging the errors in the latter stages of the development life cycle often proves very expensive. In most cases, ambiguity in understanding the requirements is the root cause. One needs to ensure that all the development efforts remain aligned toward fulfilling pre-determined requirements. When it comes to behavior-driven development vs TDD, BDD allows developers to do the above by :

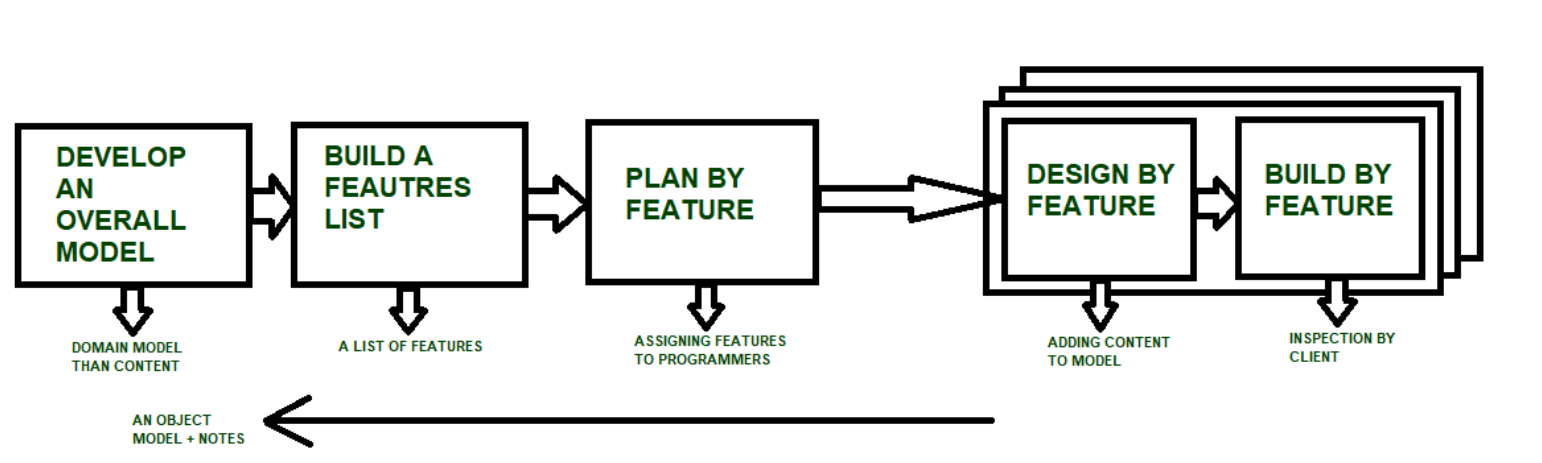
* Allowing the requirements to be defined in a standard approach using simple English
* Providing several ways to illustrate real-world scenarios for understanding requirements
* Providing a platform that enables the tech and non-tech teams to collaborate and understand the requirements

**FDD stands for Feature-Driven Development:-**

FDD stands for Feature-Driven Development. It is an agile iterative and incremental model that focuses on progressing the features of the developing software. The main motive of feature-driven development is to provide timely updated and working software to the client. In FDD, reporting and progress tracking is necessary at all levels.

FDD Lifecycle:-

* Build overall model
* Build feature list
* Plan by feature
* Design by feature
* Build by feature



Characteristics of FDD:-

* Short iterative: FDD lifecycle works in simple and short iterations to efficiently finish the work on time and gives good pace for large projects.
* Customer focused: This agile practice is totally based on inspection of each feature by client and then pushed to main build code.
* Structured and feature focused: Initial activities in lifecycle builds the domain model and features list in the beginning of timeline and more than 70% of efforts are given to last 2 activities.
* Frequent releases: Feature-driven development provides continuous releases of features in the software and retaining continuous success of the project.

Advantages of FDD :-

1. Reporting at all levels leads to easier progress tracking.
2. FDD provides continuous success for larger size of teams and projects.
3. Reduction in risks is observed as whole model and design is build in smaller segments.
4. FDD provides greater accuracy in cost estimation of the project due to feature segmentation.

Disadvantages of FDD:-

1. This agile practice is not good for smaller projects.
2. There is high dependency on lead programmers, designers and mentors

.

1. There is lack of documentation which can create an issue afterwards.