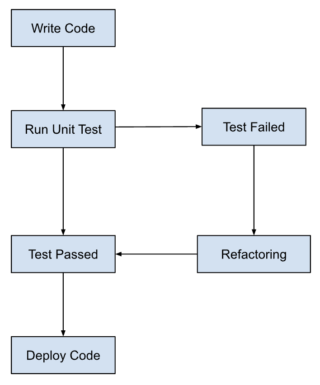
**Day3 Assignment 2: Produce a comparative infographic of TDD, BDD, and FDD methodologies. Illustrate their unique approaches, benefits, and suitability for different software development contexts. Use visuals to enhance understanding.**

**1. Test-Driven Development (TDD):**

- Approach: In TDD, developers write tests before writing the actual code, aiming to ensure that the code meets the requirements and is functioning correctly.

- Benefits: TDD helps in improving code quality, identifying errors early in the development process, and promoting test coverage.

- Suitable for: TDD is suitable for projects where requirements are well-defined and stability is important. It is also useful for projects that require frequent testing and regression testing.



**Benefits of TDD:**

**1.Ensures accurate tests and flexible codebase:**

TDD approach leads to accurate test results. Regardless of the code being implemented, TDD aims to pass the test if the business logic is robust. If any of the tests fail, TDD instantly detects the code change and informs the developers to focus on maintaining the test accuracy. It avoids the adverse effect of one unit test’s failure on another. Hence, it always ensures accurate tests. The improved test accuracy leads to the creation of a codebase that is easy to maintain and flexible. Ultimately, it reinforces the programmer’s confidence and eliminates the fear of code changes.

**2. Fosters the development of optimized code:**

TDD enables developers to effectively analyze the client’s requirements and demand clarity when they are not properly defined. The precise requirements as input will lead to the creation of optimized code.

**3. Streamlines writing documentation:**

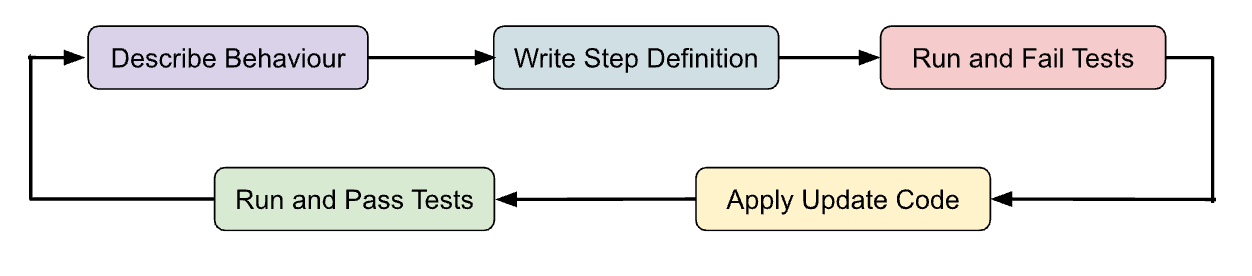
TDD allows developers to write tests for validating micro-level functionalities. Hence, it simplifies the process of writing documentation.

**2. Behaviour-Driven Development (BDD):**

- Approach: BDD focuses on the behaviour of the system and emphasizes collaboration between developers, testers, and stakeholders to define requirements in the form of scenarios.

- Benefits: BDD helps in improving communication between team members, ensuring that development is driven by business requirements, and enhancing the overall quality of the software.

- Suitable for: BDD is suitable for projects where stakeholder involvement is crucial, and there is a need for continuous communication and alignment between business goals and development activities.



**Key Benefits of BDD:**

**1. Facilitates strong collaboration:**

BDD facilitates all the involved teams to have a thorough understanding of the project. They can easily carry out profitable discussions with flawless communication. BDD allows all the involved teams to connect with the product development cycle. Since it uses easily understandable language, all the teams can effectively write behaviour scenarios.

**2. Provides high visibility:**

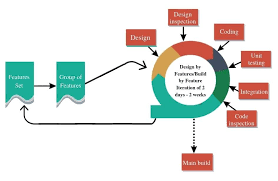
The used language is understood by all, so everybody obtains proper visibility into the project’s development. The ubiquitous language is easily understood by both technical and non-technical parties. It eliminates errors and misunderstandings within teams. Consequently, it contributes to enhanced team productivity.

**3. Cost-effective:**

BDD improves the code’s quality which leads to reduced expenses in correcting the code’s errors. Thus, it decreases the costs of maintenance and minimizes the project’s pitfalls.

**3. Feature-Driven Development (FDD):**

- Approach: FDD is an iterative and incremental software development process that focuses on building features one at a time, based on client requirement.



**Key Benefits of FDD:**

**1.Focused scope:**

By focusing on the delivery of small, incremental features, FDD helps to ensure that the development team stays focused on a well-defined scope of work, which can help to reduce the risk of scope creep and keep the project on track.

**2.Early and frequent delivery:**

FDD emphasizes the delivery of working software to the customer as quickly as possible. This allows the customer to see progress and provides the opportunity for early feedback, which can help to ensure that the final product meets their needs.

**3.Improved communication:**

FDD places a strong emphasis on communication and collaboration between team members. This can help to improve the flow of information between team members and ensure that everyone is working towards a common goal.

**4.Increased flexibility:**

FDD is an agile methodology, which means that it is designed to be flexible and responsive to changing requirements and priorities. This can help to ensure that the development team is able to adapt to changing circumstances and deliver a product that meets the customer's needs.