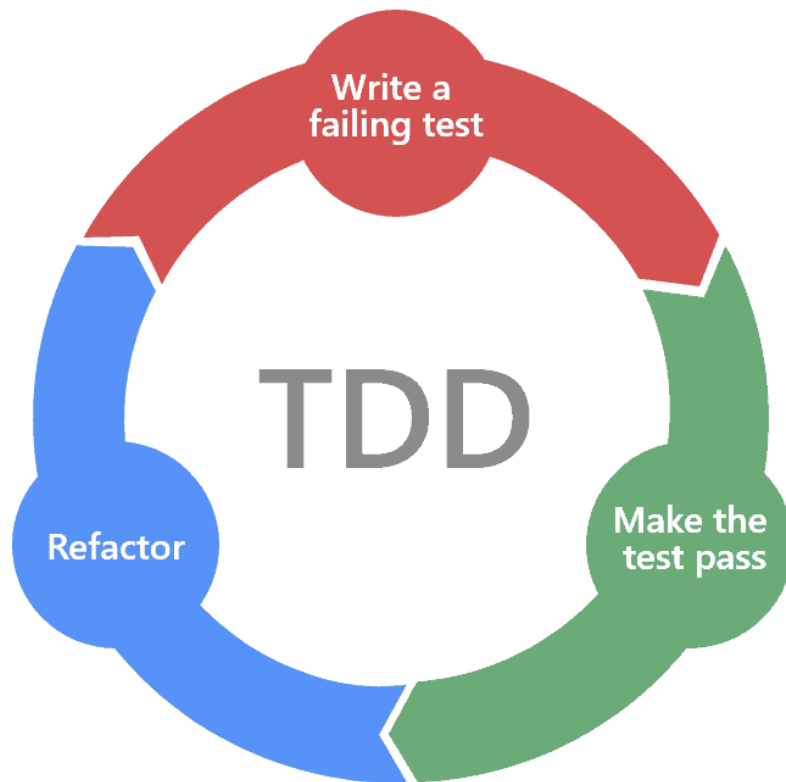


Assingment 1

1 .Create an infographic illustrating the Test-Driven Development (TDD) process. Highlight steps like writing tests before code, benefits such as bug reduction, and how it fosters software reliability.

Test-Driven Development (TDD) process



The TDD Cycle

1. Write a Test

- Developers write automated tests based on specific requirements before writing the actual code.
- Developers create test cases that define the desired behavior of the functionality to be implemented.

2. Run the Test

- Execute the test to see it fail initially.
- Tests should fail because the functionality isn't implemented yet, ensuring the test is valid.

3. Write the Code

- Write the minimum code necessary to pass the test.
- Implement code to make the failing test pass

4. Run the Test Again

- Execute the test to verify it passes.
- Ensure the newly written code passes the test, indicating successful implementation of the functionality.

5. Refactor Code

- Improve code structure without changing its functionality.
- Optimize code for readability, maintainability, and performance without altering its behavior.

6. Repeat

- Iterate through the cycle for each new feature or functionality.
- Continue writing tests, implementing code, and refactoring iteratively to build up the software.

Benefits of TDD

□ Bug Reduction

- Early detection and resolution of bugs due to continuous testing.
- Tests Catch issues before the code is deployed

□ Software Reliability

- Fosters overall software reliability by continually verifying functionality.
- Confidence in the code stability and reliability.

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.

Creating a comparative infographic for Test-Driven Development (TDD), Behavior-Driven Development (BDD), and Feature-Driven Development (FDD) involves outlining each methodology's unique approach, benefits, and suitability for different software development contexts. Here's a detailed textual plan for the infographic, which can be later translated into a visual design.

Comparing Software Development Methodologies: TDD, BDD, and FDD

Overview of Methodologies

1. Test-Driven Development (TDD)

- **Approach:** Write tests before code.
- **Benefits:** Reduces bugs, improves code quality, ensures code meets requirements.
- **Suitability:** Ideal for small to medium-sized projects where unit testing is crucial.

2. Behavior-Driven Development (BDD)

- **Approach:** Write tests in plain language based on user behavior.
- **Benefits:** Enhances communication, ensures user needs are met, improves collaboration.
- **Suitability:** Suitable for projects with complex requirements needing clear communication

between stakeholders.

3. Feature-Driven Development (FDD)

- **Approach:** Develop features iteratively and incrementally.
- **Benefits:** Provides frequent tangible progress, easy to track, and scalable.
- **Suitability:** Best for large-scale projects with a need for constant progress visibility.

Comparative Table

□ Approach

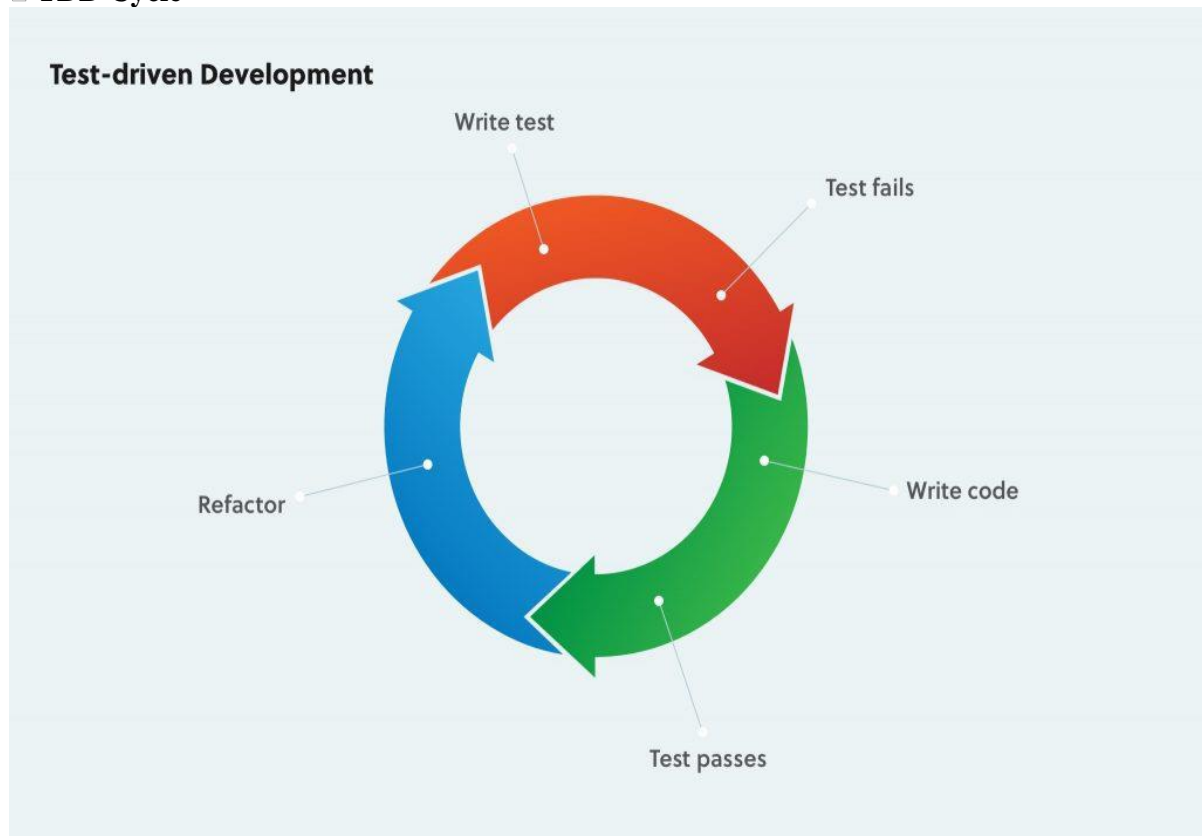
- TDD: Write tests before code.
- BDD: Write tests based on user behavior.
- FDD: Develop features iteratively.

□ Benefits

- TDD: Bug reduction, improved code quality.
- BDD: Better communication, meets user needs.
- FDD: Tangible progress, easy tracking.
- **Suitability**
 - TDD: Small to medium projects.
 - BDD: Complex projects with diverse stakeholders.
 - FDD: Large-scale projects.

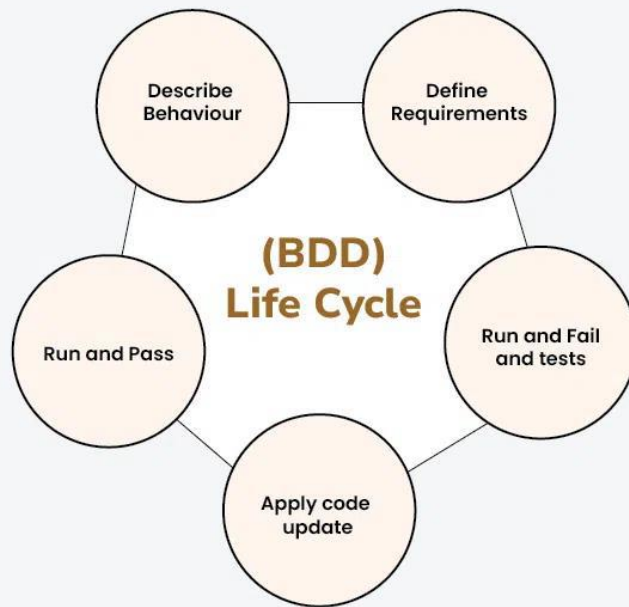
Visual Representation

□ TDD Cycle



BDD Cycle

Behavior-Driven Development (BDD) Life Cycle



FDD Cycle

FDD – 5 Stage Processes

