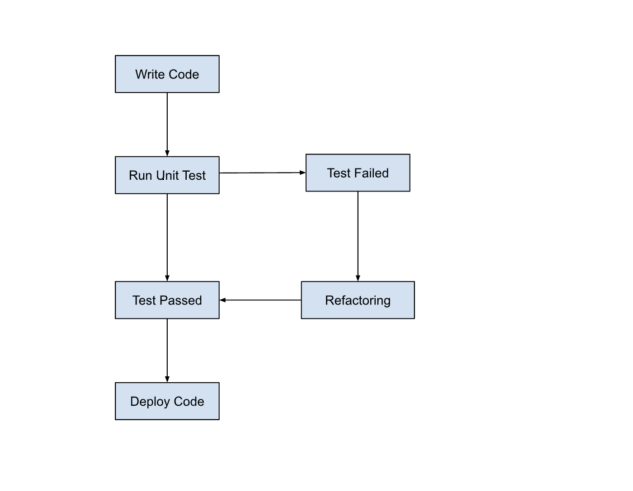
**Assignment 1: Create an infographic illustrating the Test-Driven Development (TDD) process. Highlight steps like writing tests before code, benefits such as bug reduction, and how it fosters software reliability**

1. Write a Test

Write Test: Create a test case for the defined behavior. It should fail initially.

2. Run the Test

Execute: Run the test to confirm it fails.

3. Write Code

Implement: Write the simplest code to pass the test.

4. Run the Test Again

Execute: Run all tests including the new one. It should pass now.

5. Refactor Code

Cleanup: Improve the code structure without altering its behavior.

Maintainability: Ensure code is easy to understand and maintain.

Benefits of TDD:

• Early Bug Detection: Identify issues at the earliest stage.

• Improved Code Quality: Encourages modular, maintainable code.

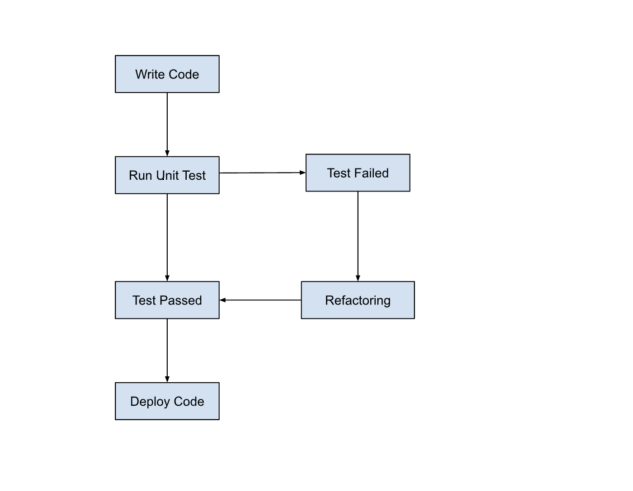
• Enhanced Design: Focuses on requirements, leading to better software architecture.

• Reduced Debugging Time: Less time spent on debugging due to early error detection.

• Better Documentation: Tests serve as living documentation of the system's behavior.

**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:**

Write a failing test.

Write the code to pass the test.

Refactor the code.

**Benefits:**

Promotes cleaner and modular code.

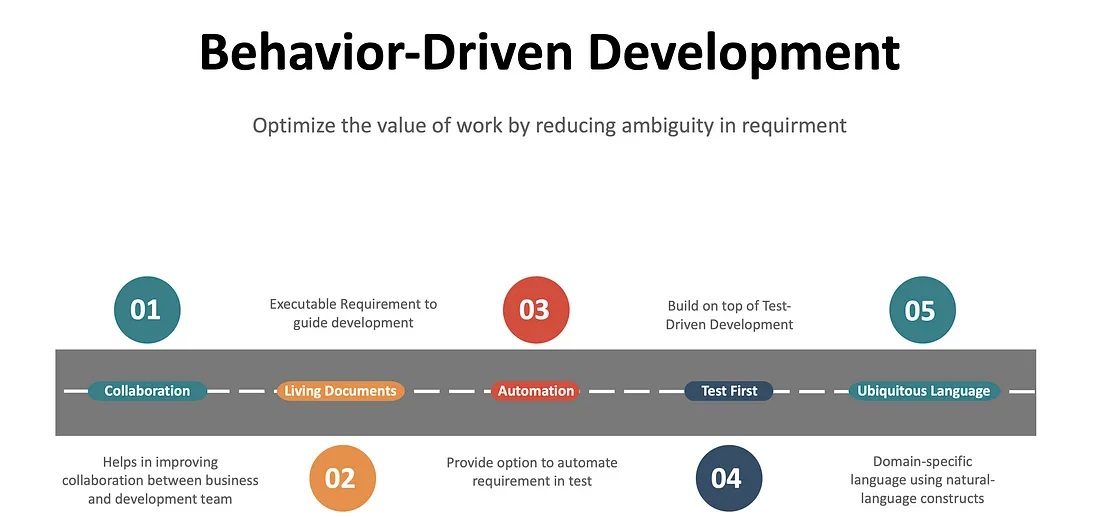
Early detection of bugs.

**Suitability:**

Suitable for projects with clear requirements.

Works well in Agile environments.

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

****

**Approach:**

Define behavior using the Given-When-Then format.

Implement code to fulfill the behavior.

Validate behavior through tests.

**Benefits:**

Encourages collaboration between developers, QA, and stakeholders.

Focuses on user stories and business outcomes.

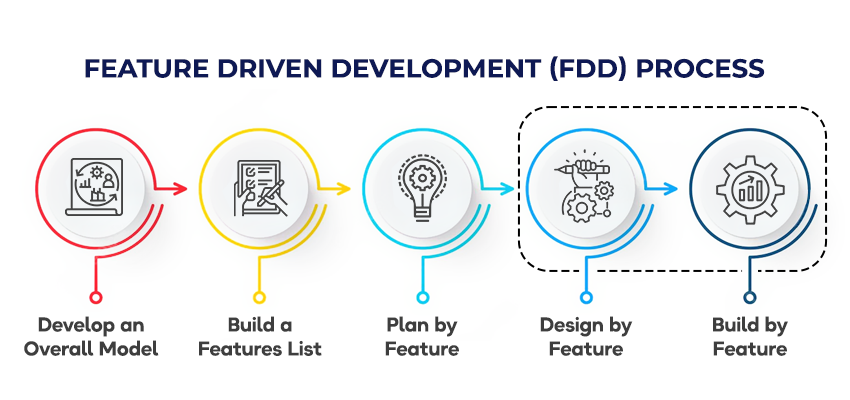
Improves communication and understanding of requirements.

**Suitability:**

Ideal for projects with complex business logic.

Suitable for cross-functional teams.

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

****

**Approach:**

Develop an overall model.

Build a feature list.

Plan by feature, design by feature, and build by feature.

**Benefits:**

Facilitates clear project management and progress tracking.

Scales well to large projects.

**Suitability:**

Best suited for large, complex projects.

Works well in environments requiring a structured approach.

**Conclusion:**

Each methodology offers distinct approaches and benefits.

Choose the methodology based on project size, complexity, and team structure.

