ASSIGNMENT - MODERN DEVELOPMENT METHODOLOGIES

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.**

Steps in the TDD Process

Step 1 : Write a Test

The process begins with writing a test for a specific functionality before writing the actual code. This test is expected to fail initially because the functionality it tests does not exist yet.

Step 2 : Run the Test

The next step is to run the test, which should fail as expected. This confirms that the test is working correctly and that it fails for the right reason.

Step 3 : Write the Code

Now, write the minimum amount of code necessary to pass the test. The goal here is not to write perfect code, but to write code that makes the test pass.

Step 4 : Run Tests

Run all tests to ensure that the new code did not break any existing functionality and that the new functionality works as expected.

Step 5 : Refactor Code

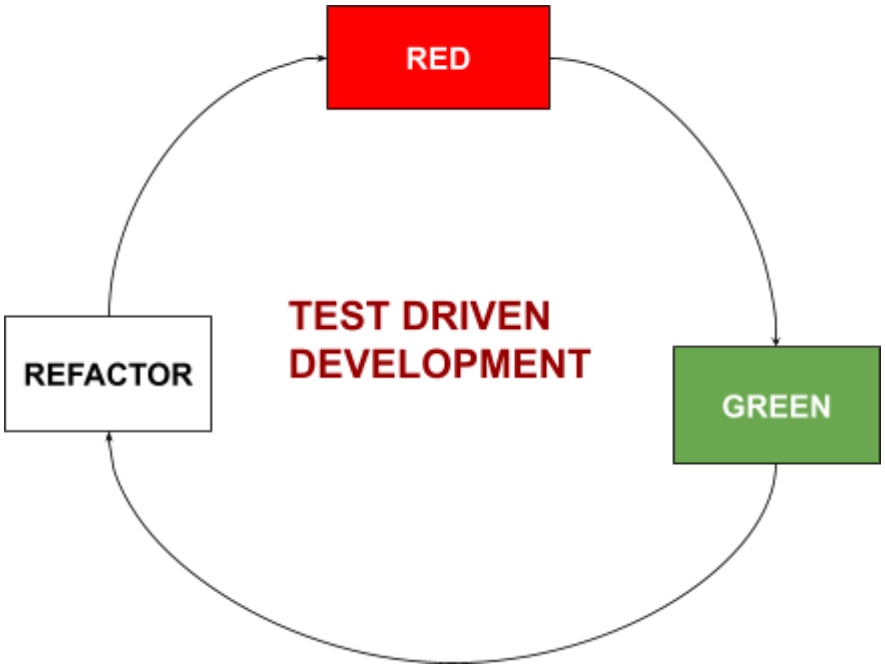
The final step is to clean up the code, removing any duplication and improving its structure, while ensuring that all tests still pass.

Benefits of TDD

- **Bug Reduction:** TDD helps in reducing the number of bugs in the software. By writing tests first, developers ensure that the code works as expected before it is integrated into the system.
- **Software Reliability:** TDD fosters software reliability by ensuring that all code is tested and works as expected. This makes the software more robust and less prone to errors.
- **Improved Design:** TDD encourages developers to think about the design of their code before they write it, leading to better-structured and more maintainable code.

- **Easier Maintenance:** With a suite of tests for every piece of functionality, it's easier to maintain and refactor the code without fear of breaking existing functionality.

Infographic illustrating the TDD process



Red – Create a test case and make it fail.
Green – Make the test case pass by any means.
Refactor – Change the code to remove duplicate/redundancy.

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.

(i) TDD (Test-Driven Development):

Approach:

- Visual representation of a developer writing a failing test.
- Caption: "Write failing tests before writing code to drive development."

Benefits:

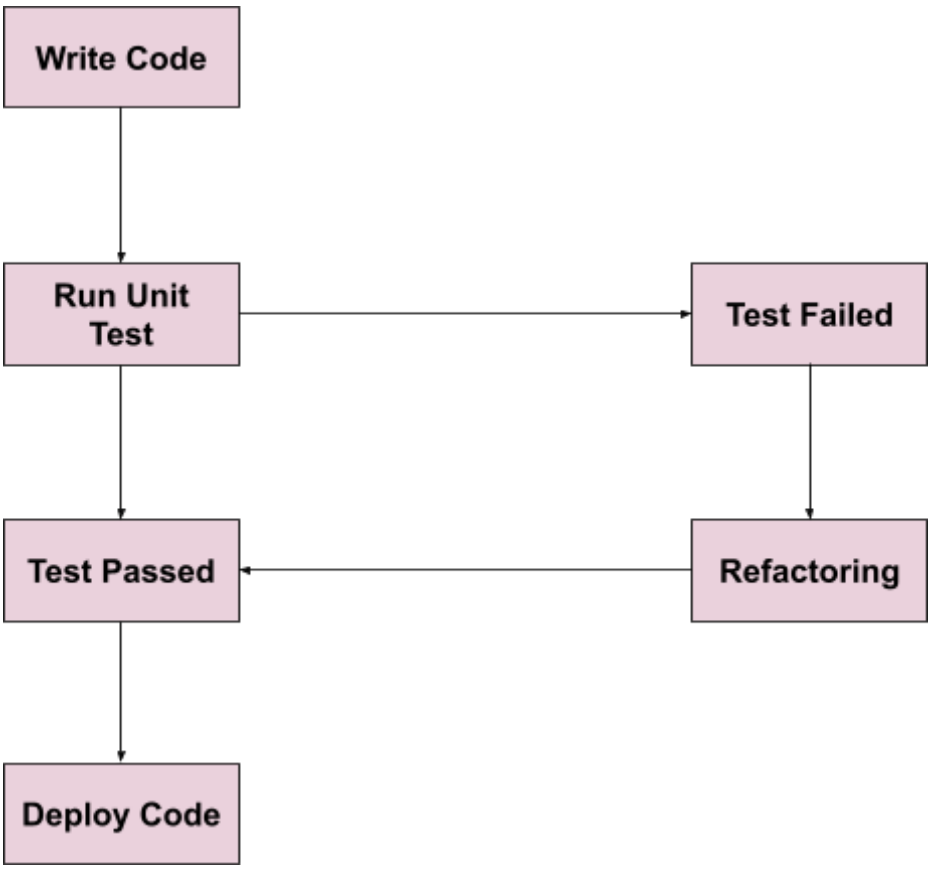
- Visual representation of a scale tipping in favor of TDD.
- Bulleted list of benefits, including:
- Early bug detection.

- Improved code quality.
- Enhanced software reliability.
- Faster development cycles.

Suitability:

- Visual representation of TDD fitting into a puzzle piece labeled "Iterative Development."
- Caption: "Best suited for iterative development processes and projects requiring frequent changes."

Refactoring refers to modifying the code without changing its main functionality or behavior



(ii) BDD (Behavior-Driven Development):

Approach:

- Visual representation of a team discussing user stories and behaviors.
- Caption: "Define behavior through scenarios and acceptance criteria before writing code."

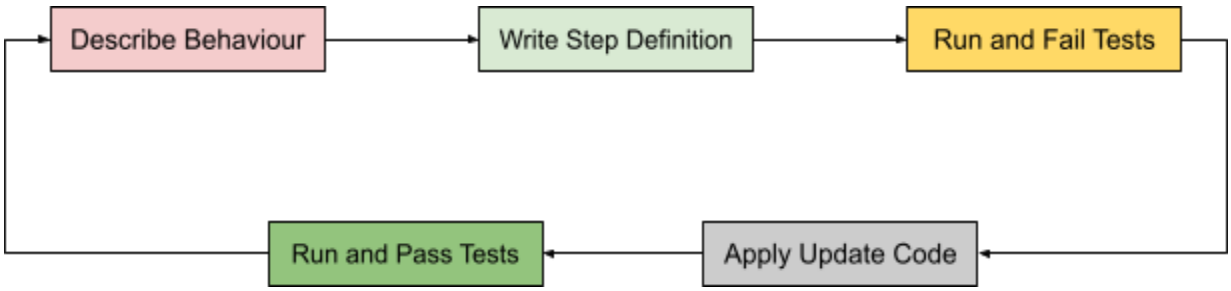
Benefits:

- Visual representation of a scale tipping in favor of BDD.
- Bulleted list of benefits, including:
- Improved collaboration between stakeholders.
- Focus on business requirements.
- Enhanced clarity and understanding of features.
- Better alignment between development and business goals.

Suitability:

- Visual representation of BDD fitting into a puzzle piece labeled "Collaborative Environments."
- Caption: "Best suited for projects with complex business logic and involving multiple stakeholders."

The image below depicts a typical BDD operation:



(iii) FDD (Feature-Driven Development):

Approach:

- Visual representation of breaking down features into manageable chunks.
- Caption: "Focus on delivering features iteratively based on client priorities." Benefits:
- Visual representation of a scale tipping in favor of FDD.
- Bulleted list of benefits, including:
- Clear focus on features and deliverables.
- Improved project visibility and progress tracking.
- Efficient resource allocation.
- Enhanced client satisfaction.

Suitability:

- Visual representation of FDD fitting into a puzzle piece labeled "Client-Centric Projects."
- Caption: "Best suited for projects with well-defined requirements and a focus on delivering features based on client needs."