Assignment-3

Assignment1: 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.

Solution:-

Section 1: What is TDD?

Definition: A software development approach where tests are written before the actual code.
Purpose: Ensure the code works as expected and improve overall quality.

Section 2: TDD Cycle

Write a Test:
Begin by writing a test for a new feature or functionality.
Ensure the test is specific and fails initially.

Run the Test:
Execute the test to see it fail.
This step confirms the test is valid and the feature is not yet implemented.

Write Code:
Write the minimum code required to make the test pass.
Focus on simple and direct solutions.

Run Tests Again:
Execute all tests to ensure the new code passes the test.
Verify no other tests are broken.

Refactor:
Improve the code structure while keeping the functionality intact.
Clean up any redundancy or inefficiencies.

Repeat:
Continue this cycle for each new feature or bug fix.

Section 3: Benefits of TDD

Bug Reduction:
Early detection and fixing of bugs.
Reduces the chances of defects in production.

Software Reliability:
Continuous testing ensures stable and reliable code.
Helps maintain a high level of confidence in the software.

Documentation:
Tests serve as documentation for the code.
Easier for new developers to understand the functionality.

Design Improvement:
Encourages better design and modularity.
Facilitates cleaner and more maintainable code.

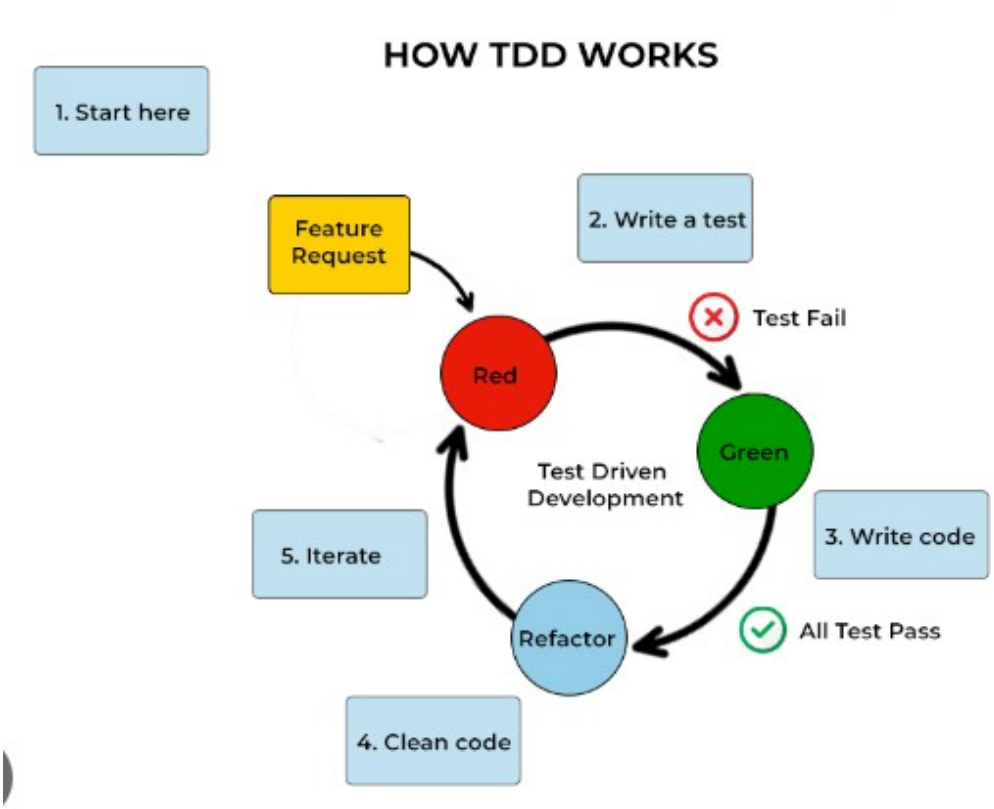
Cost-Effective:
Reduces the cost of fixing bugs later in the development cycle.
Saves time and resources in the long run.

Section 4: Visual Representation of the TDD Cycle

Flowchart:
Start → Write a Test → Run the Test (Fail) → Write Code → Run Tests (Pass) → Refactor → Repeat

Use arrows to connect each step, creating a loop that highlights the iterative nature of TDD.

Section 5: Conclusion
Summary:
TDD is a powerful methodology for enhancing software quality.
By incorporating TDD, developers can ensure robust, reliable, and maintainable code.
Call to Action:
Encourage developers to adopt TDD for better software development practices.



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.

Solution:-

Overview:-

TDD (Test-Driven Development)
Process: Write tests first, then code.
Benefits: Reduces bugs, improves code quality.

BDD (Behavior-Driven Development)
Process: Define behavior, write tests, then code.
Benefits: Enhances communication, focuses on user needs.

FDD (Feature-Driven Development)
Process: Plan, design, and build by feature.
Benefits: Manages large projects well, delivers features regularly.

Process Diagrams:-

TDD Cycle:
Write Test → Run Test (Fail) → Write Code → Run Tests (Pass) → Refactor → Repeat

BDD Cycle:
Define Behavior → Write Scenario → Write Test → Develop Code → Run Test (Pass) → Repeat

FDD Cycle:
Identify Features → Plan → Design → Build → Inspect → Deploy → Repeat

Benefits:-

TDD:
Early bug detection, high code quality.

BDD:
Better stakeholder communication, user-focused development.

FDD:
Handles large projects, regular feature delivery.

Best For:-

TDD:
Critical, high-reliability projects.

BDD:
Projects needing strong stakeholder involvement.

FDD:
Large, feature-rich projects.