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.

Creating an infographic for Test-Driven Development (TDD) involves visually outlining the key steps and highlighting the benefits. Here's a detailed description to help you design this infographic:

Infographic Title: *Test-Driven Development (TDD) Process*

Section 1: Introduction to TDD

Header: What is Test-Driven Development?

Text: TDD is a software development process where tests are written before the actual code. This approach ensures that the code is working correctly from the start.

Section 2: TDD Process Steps

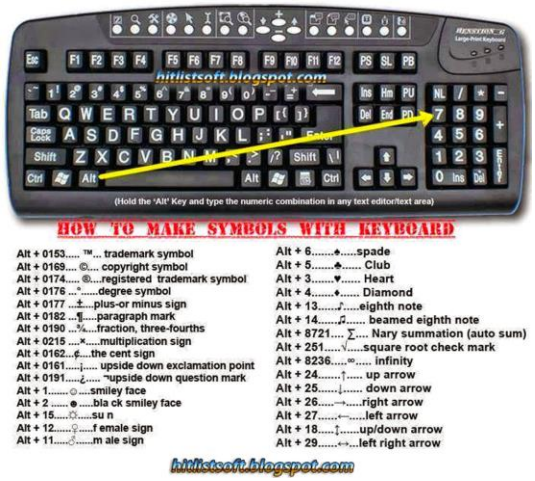
Write a Test:

Text: Write a test for a new feature or function. The test should be specific and cover all expected outcomes.

Run the Test:

Text: Execute the test. At this stage, it should fail because the feature isn't implemented yet.

Write the Code:



Text: Write the minimum amount of code necessary to pass the test.

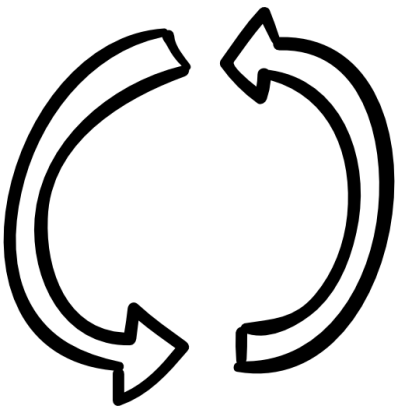
Run All Tests:

Text: Execute all tests to ensure the new code doesn't break existing functionality.

Refactor Code:

Text: Improve and optimize the code while ensuring all tests still pass.

Repeat:



Text: Continue the cycle for each new feature or function.

Section 3: Benefits of TDD

Bug Reduction:

Text: Writing tests first helps identify and fix bugs early in the development process.

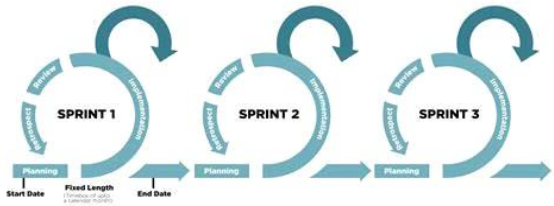
Fosters Software Reliability:

Text: Continuous testing ensures that the software behaves as expected and remains stable over time.

Improves Code Quality:

Text: Refactoring leads to cleaner, more maintainable code.

Facilitates Agile Development:



Text: TDD supports iterative development and quick adaptation to changes.

Section 4: Visual Summary (Flowchart)

Flowchart Layout: Arrange the steps in a circular or linear flowchart to visually represent the iterative nature of TDD.

Start with "Write a Test" → "Run the Test" → "Write the Code" → "Run All Tests" → "Refactor Code" → Back to "Write a Test"

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.

To create a comparative infographic for Test-Driven Development (TDD), Behavior-Driven Development (BDD), and Feature-Driven Development (FDD), you can follow this detailed plan:

Infographic Title: *Comparing Software Development Methodologies: TDD, BDD, and FDD*

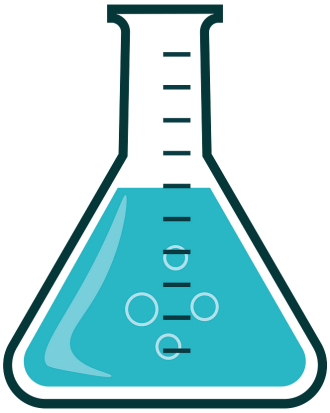
Section 1: Introduction

Header: Understanding Development Methodologies

Text: A comparison of TDD, BDD, and FDD to help choose the best approach for your software project.

Section 2: Overview of Methodologies

Test-Driven Development (TDD):



Text: TDD is a process where tests are written before the actual code to ensure functionality.

Behavior-Driven Development (BDD):

Text: BDD extends TDD by writing test cases in natural language to describe behavior and collaboration.

Feature-Driven Development (FDD):

Text: FDD focuses on building features guided by client-valued functionality, with short iteration cycles.

Section 3: Process Comparison

TDD Process Steps:

1. Write a Test
2. Run the Test

3. Write the Code
4. Run All Tests
5. Refactor Code
6. Repeat
7. Visual: Circular flowchart

BDD Process Steps:

1. Define Behavior in User Stories
2. Write Scenarios in Gherkin Language
3. Develop Step Definitions
4. Write the Code to Pass Scenarios
5. Run Tests
6. Refactor
7. Visual: Linear flowchart with speech bubbles

FDD Process Steps :

1. Develop Overall Model
2. Build Feature List
3. Plan by Feature
4. Design by Feature
5. Build by Feature
6. Visual: Sequential flowchart

Section 4: Benefits Comparison

TDD Benefits:

- Reduces bugs early
- Ensures reliability
- Improves code quality

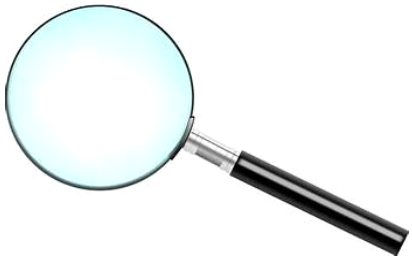
BDD Benefits:

- Enhances collaboration between developers and non-developers
- Ensures software meets user expectations
- Creates clear documentation

FDD Benefits:

- Accelerates development with short iterations
- Focuses on client-valued features
- Can be cost-effective by reducing wasted effort
- Section 5: Suitability for Different Contexts

TDD:



Text: Suitable for projects needing high reliability and maintainability, often used in high-stakes environments like finance or healthcare.

BDD:

Text: Ideal for projects requiring close collaboration with stakeholders and clear communication of requirements, like web applications.

FDD:

Text: Best for larger projects with well-defined requirements, where quick delivery of features is essential, often seen in enterprise applications.