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.

Test-Driven Development (TDD) Process :

Test-Driven Development (TDD) is a software development approach where tests are written before the actual code. The TDD cycle involves writing a test, running the test (which will initially fail), writing the minimal code needed to pass the test, and then refactoring the code while ensuring that all tests still pass. This process is repeated iteratively.

Steps in TDD :

1. **Write a Test :** Before writing any code, write a test for the functionality you want to implement.
2. **Run the Test :** Run the test to see it fail. This step ensures the test is valid and the functionality isn't already present.
3. **Write Code :** Write the minimum amount of code necessary to make the test pass.
4. **Run the Test Again :** Run the test again to see it pass. This step confirms the new code meets the test's requirements.
5. **Refactor Code :** Refactor the code to improve its structure and readability without changing its behavior.
6. **Repeat :** Repeat the cycle for each new functionality.

Benefits of TDD

1. **Bug Reduction :** Writing tests before code helps in identifying and fixing bugs early in the development process.
2. **Improved Code Quality :** Regular refactoring leads to cleaner, more maintainable code.
3. **Software Reliability :** Ensures that new changes do not break existing functionality, fostering reliable software.
4. **Documentation :** Tests serve as documentation for the code, making it easier to understand the functionality.
5. **Continuous Feedback :** Provides immediate feedback on the code’s correctness, allowing for quick adjustments.

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.

Test-Driven Development (TDD) :

Approach:

- Write tests before writing the actual code.
- Follow the cycle: Write a test -> Run the test (fail) -> Write code -> Run the test (pass) -> Refactor.

Benefits:

- Ensures code is thoroughly tested.
- Helps in catching bugs early.
- Encourages simple and clean code.
- Acts as documentation for the code.

Suitability:

- Suitable for projects where code quality and reliability are critical.
- Best for environments that encourage regular refactoring and incremental development.

Behavior-Driven Development (BDD) :

Approach:

- Extends TDD by writing test cases in natural language.
- Involves collaboration between developers, testers, and non-technical stakeholders.
- Uses Gherkin language (Given-When-Then) to define tests.

Benefits:

- Improves communication between technical and non-technical team members.
- Ensures that all stakeholders understand the requirements.
- Facilitates test automation and continuous integration.

Suitability:

- Ideal for projects with frequent changes in requirements.
- Suitable for agile environments where collaboration and understanding of user behavior are crucial.

Feature-Driven Development (FDD) :

Approach:

- Emphasizes building and designing features.
- Follows a five-step process: Develop an overall model -> Build a features list -> Plan by feature -> Design by feature -> Build by feature.

Benefits:

- Focuses on delivering tangible, working features.
- Encourages iterative and incremental development.
- Provides clear progress tracking and reporting.

Suitability:

- Best for large-scale projects with complex requirements.
- Suitable for teams that prefer a structured approach to feature development.

Assignment 1 :

Agile Project Planning - Create a one-page project plan for a new software feature using Agile planning techniques. Include backlog items with estimated story points and a prioritized list of user stories.

Payment feature for Amazon

Features:

1: Payment Methods Management

2: Payment Confirmation

3: One-Click Payment

4: Promotional Code Application

User Stories and Tasks:

Feature 1: Payment Methods Management

User Story 1: As a user, I want to save multiple payment methods securely.

Story Points: 8

Tasks:

Building the UI- 16hr

Building the Middle Layers – 8 hrs.

Building the Database Layers- 4hrs.

User Story 2: As a user, I want to set a default payment method.

Story Points: 5

Tasks:

Building the UI- 16hr

Building the Middle Layers – 8 hrs.

Building the Database Layers- 4hrs.

User Story 3: As a user, I want to see a summary of my payment methods.

Story Points: 5

Tasks:

Building the UI- 16hr

Building the Middle Layers – 8 hrs.

Building the Database Layers- 4hrs.

Feature 2: Payment Confirmation

User Story 4: As a user, I want to receive a confirmation email after a successful payment.

Story Points: 3

Tasks:

Building the UI- 16hr

Building the Middle Layers – 8 hrs.

Building the Database Layers- 4hrs.

Feature 3: One-Click Payment

User Story 5: As a user, I want to use a one-click payment option.

Story Points: 8

Tasks:

Building the UI- 16hr

Building the Middle Layers – 8 hrs.

Building the Database Layers- 4hrs.

Feature 4: Promotional Code Application

User Story 6: As a user, I want to apply promotional codes during checkout.

Story Points: 3

Tasks:

Building the UI- 16hr

Building the Middle Layers – 8 hrs.

Building the Database Layers- 4hrs.

Prioritized of User Stories:

User Story 5: One-Click Payment (8 SP)

User Story 1: Save Multiple Payment Methods (8 SP)

User Story 2: Set Default Payment Method (5 SP)

User Story 3: Payment Methods Summary (5 SP)

User Story 4: Payment Confirmation Email (3 SP)

User Story 6: Apply Promotional Codes (3 SP)

Assignment 2 :

Daily Standup Simulation - Write a script for a Daily Standup meeting for a development team working on the software feature from Assignment 1. Address a common challenge and incorporate a solution into the communication flow.

The development team address a common challenge and incorporate a solution payment feature, especially with the bank gateways (SBI, AXIS, HDFC, CANARA ,...)

Challenge -
We encountered a recurring issue where the bank didn't respond to our requests, happening 20 times transaction attempts, out of 18 are successfully transacted while 2 fail due to the bank not responding. This problem slows down our progress significantly, with 18 instances of this happening yesterday alone. To tackle this.

Ssolution -
Decide to implement a retry logic ,Now, every failed request will automatically
be retried up to three times, ensuring we don't get stuck