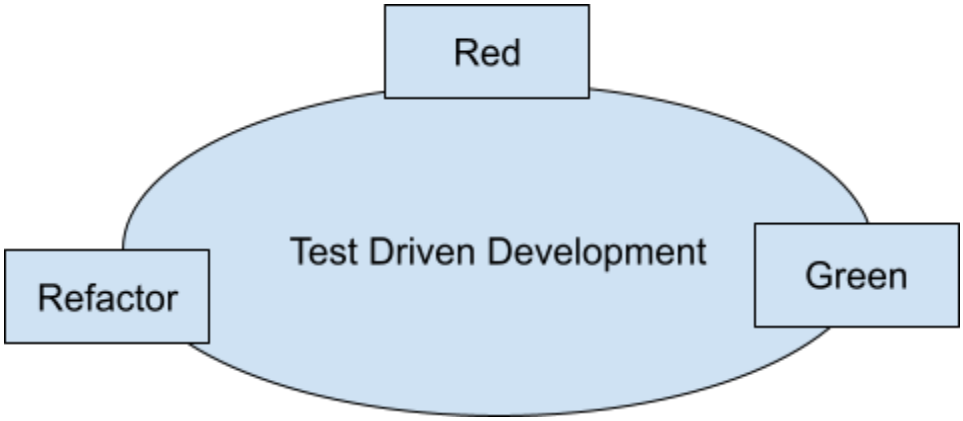
Assignment 1: Create an infographic illustrating the Test-Driven Development (TDD) process. Highlight steps like writing tests before code, benefits such as bug reduction, and how it fosters software reliability.

Test Driven Development is the process in which test cases are written before the code that validates those cases. It depends on repetition of a very short development cycle. Test driven Development is a technique in which automated Unit test are used to drive the design and free decoupling of dependency.



In above diagram:

Red - Create a test case and make it fail.
Green - Create a test case pass by any means.
Refactor - Change the code to remove duplicates.

- ➢ Unit tests provide constant feedback about the functions.
- ➢ Quality of design increases which further helps in proper maintenance.
- ➢ Test driven development acts as a safety net against the bugs.
- ➢ TDD ensures that your application actually meets requirements defined for it
- ➢ TDD has a very short development lifecycle.

Advantages:

- TDD can enable faster innovation and continues the delivery
- The result code is easy to test.
- TDD is flexible and extensible

Disadvantages:

- Forget to test the code frequently.
- Write too many tests at once.
- Fails to maintain the tests.

Task 1: Write the differences between the Test-Driven Development and Behaviour Driven Development.

Test Driven Development:

- Test Driven Development is a development technique which focuses more on the implementation of a feature of a software application/product
- In TDD the participants are developers.
- Its main focus is on unit tests.
- In TDD the starting point is a test case.
- It is a development practice.
- In TDD collaboration is required only between the developers
- It is a good approach for projects which involve API and third-party tools.
- Some of the tools used are Cucumber,Jbehave,and Jdave.

Behavior Driven Development:

- Behavior Driven Development is a development technique which focuses more on a software application's behavior.
- In BDD the participants are Developers, Customers, QAs.
- Its main focus is on system requirements.
- In BDD the starting point is a scenario.
- It is a team methodology.
- In BDD collaboration is required between all the stakeholders

- It is a good approach for project development which is driven by user actions.
- Some of the tools are Cucumber,Jbehave and Jspec.

Task 2:Write the Feature scenario for Login and transaction

Scenario 1: Successful Login

- Given a registered user
- Enter valid username and password
- It will successfully logged into the system
- Then it should be redirected to the dashboard

Scenario 2: Invalid Login

- Given a registered user
- Enter an invalid username or password
- Then we will see an error message indicating the credentials are incorrect
- And it should remain in the Login page.

Scenario 3: Forgot Password

- Given a registered user
- Click on the "Forgot Password" link
- Then it will be directed to a page to reset my password
- Then we will receive an email to reset the password

Scenario 4: View Account Balance

- Logged into the system

- It will navigate to the account dashboard
- Then we are able to view current account balance

Scenario 5: Make a Transaction

- Logged into the system
- A transaction (e.g., transfer funds, make a payment)
- Then we are able to select the recipient and specify the amount
- And we will receive a confirmation message upon successful transaction completion

Scenario 6: Insufficient Funds

- Logged into the system
- Insufficient funds to complete a transaction
- Then we attempt to make a transaction
- Then it we will receive an error message indicating insufficient funds
- And the transaction should not be processed

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.

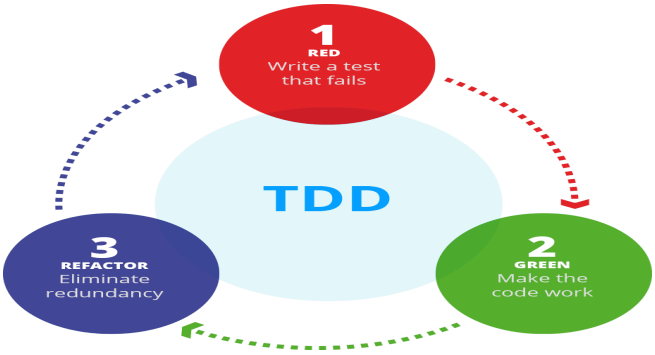
1. **Test-Driven Development (TDD):**

Approach: Write tests before writing code.

Benefits:

- ➢ Ensures code meets requirements.
- ➢ Encourages modular and maintainable code.
- ➢ Provides a safety net for refactoring.

Suitability: Best for small to medium-sized projects. Ideal when requirements are clear.



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

   Approach: Define behavior in natural language specifications.

   Benefits:

- ➢ Promotes collaboration between developers, QA, and stakeholders.
- ➢ Enhances understanding of requirements.
- ➢ Produces living documentation.

Suitability: Suitable for projects with complex business logic or when working closely with non-technical stakeholders.

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

Approach: Breaks down development into features.

Benefits:

- ➢ Emphasizes on delivering features quickly.
- ➢ Promotes a structured development process.
- ➢ Encourages team collaboration and accountability.

Suitability: Well-suited for large-scale projects with multiple teams. Works best when requirements are continually evolving.

Task 3: Difference between the Agile methodology and SDLC processes.

Software Development Lifecycle:

1.SDLC (Software Development Lifecycle) is a systematic process used for efficient project management
2.SDLC approach is Linear and sequential development process
3.SDLC has different phases: Planning, Design, Coding, Testing, Deployment, Maintenance.
4.Example frameworks:Waterfall model,Spiral model, V-model
5.Feedback is typically gathered at the end of each phase
6.Full product is delivered at the end of the development cycle.

Agile Methodology:

1.Agile is an iterative approach and methodology embedded within the SDLC for Software Project Development
2.Agile method approach is Iterative and incremental development.
3.Agile method has phases which are divided into short, time-boxed iterations.
4.Example frameworks:Scrum, Kanban, Extreme Programming (XP)
5.Feedback is frequent and continuously taken from the stakeholders
6.Allows for the delivery of a Minimum Viable Product (MVP) early in the process