K DEEPAK **ASSIGNMENT(DAY-3)**

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:

Write test cases : Developers write automated test cases based on requirements before introducing any production code.

Run tests : Excution of the tests cases in to ensure there fail initialy, introduce and indicating that the tests very effective.

code:Developing the minimum code to applicable necessary overcome the tests.

Run tests again:Executing the tests again to validate that the new passes all the tests.

Refactor the code: Optimize the refactor the code while ensuring that all the tests continue to pass

Benefits of TDD:

Bug Reduction: By catching bugs early in the development process, TDD reduces the number of defects in the final product.

Improve Software Reliability: Rigorous testing throughout development leads to more robust and reliable software.

Better Code Design: TDD encourages modular and loosely coupled code, leading to easier maintenance and scalability.

Faster Development: Despite the initial investment in writing tests, TDD can accelerate development by streamlining debugging and reducing rework.

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 visual to enchance understanding.

Analysis of software Development Methodologies:

Software development Is representation of the three methods TDD, BDD, and FDD.

Test Driven Development:

Approach:

The coders or developers write the test cases Before writing the Main code . which we get the logical approach of the code or design of the code

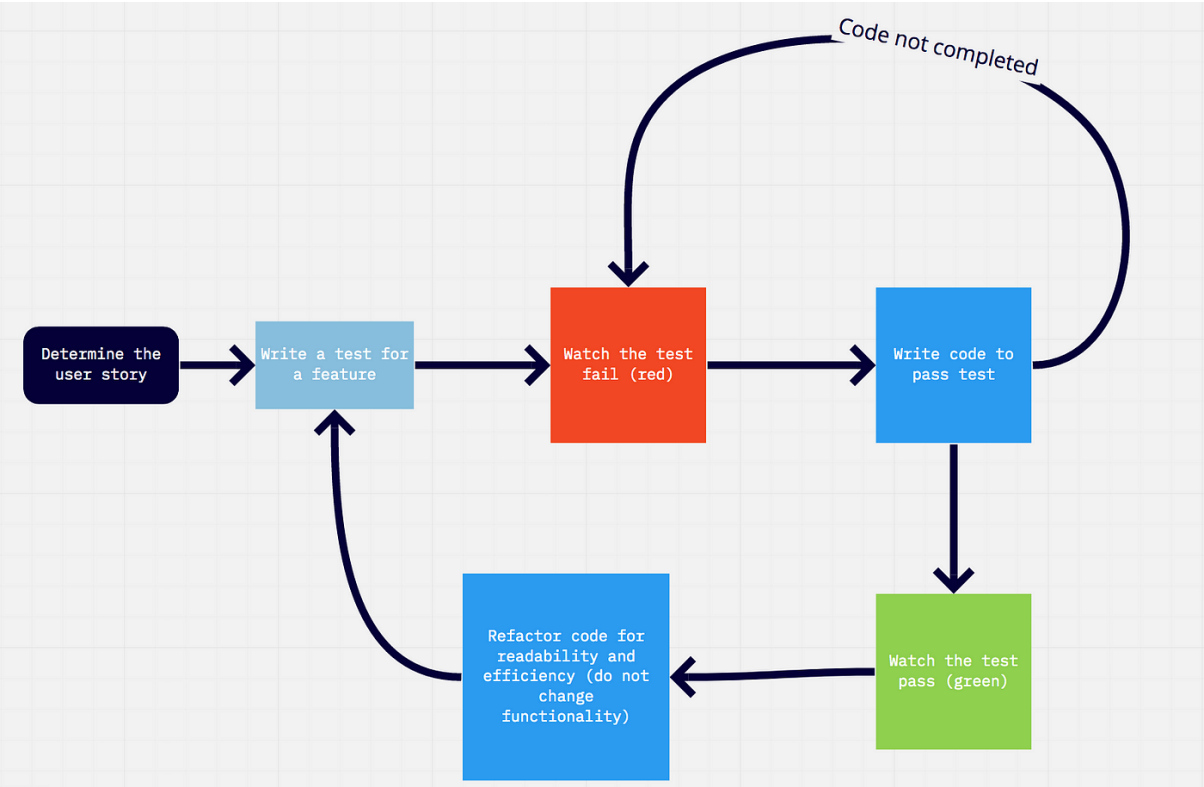
Benefit :

- The team will get the error of the pre written code and can change the early time period.
- The TDD encourage the modules or modular and it will done by the loose coupling method
- It useful for the maintence of the code durability and code designing.

Validation:

- Projects get benefited by the requirements are cleared and stabilized.
- Suitable for projects with a focus on code quality and reliability.

Flowchart of TDD:



Feature Driven Development(FDD):

Approach:

- Develop an overall model of the SDLC in agile method and build the feature list based on the model.
- It has plan ,design,and build features itertively.
- Inspectand adapt continuously to progress and quality.

Benefit:

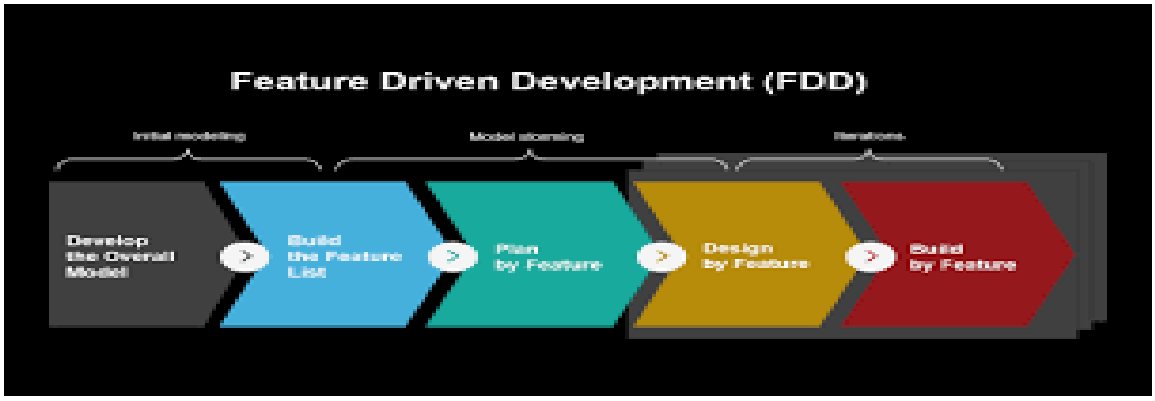
- Identify and prioritize features based on user requirements and business priorities. Break down features into smaller tasks, estimate effort.
- plan iterations to implement them. Design and implement each feature incrementally, focusing on delivering working functionality.
- Develop and test each feature independently before integrating it into the system. Iterate through the process for each feature until all planned features are implemented.

Validation:

- It is Suitable for large-scale projects with diverse feature sets.

- The team has the Ideal for teams working in a collaborative and iterative manner.

Flowchart of TDD:



Behaviour Driven Development(FDD):

Approach: Behavior-Driven Development (BDD) is an agile approach to software development centered around expressing stakeholder expectations in everyday language through clear and understandable specifications.

Benefits :

- Promotes effective communication and teamwork among developers, QA teams, and non-technical stakeholders through a shared language.

- Ensures that development activities are in line with business objectives and user needs.

- Advocates for a test-driven approach akin to TDD but emphasizes behavioral expectations over detailed unit tests.

Validation:

- Outline user requirements or features using natural language guidelines (e.g., Given-When-Then)

- . Draft executable specifications (BDD tests) based on these guidelines using tools like Cucumber, SpecFlow, or Behave.

- Code implementation to fulfill the specifications, ensuring each behavior is accurately addressed.

- Execute BDD tests to confirm the implemented code fulfills the expected behaviors.

Flowchart of BDD: