Assignment-3

Name : Joshnitha Rangolu

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 dependencies.

The following sequence of steps is generally followed:

1.Add a test – Write a test case that describe the function completely. In order to make the test cases the developer must understand the features and requirements using user stories and use cases.

2.Run all the test cases and make sure that the new test case fails.

3.Write the code that passes the test case

4.Run the test cases

5.Refactor code – This is done to remove duplication of code.

6.Repeat the above mentioned steps again and again

Benefits of TDD

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

How TDD Fosters Software Reliability
- Tests act as a safety net for the codebase
- Regression testing with each code change
- Encourages modular and testable code design
- Facilitates continuous integration and delivery
- Enables refactoring and code maintenance with confidence Challenges and Best

Practices

- Initial learning curve and mindset shift
- Writing good tests (FIRST principles)
- Test code organization and maintenance
- Balancing TDD with other development approaches

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.

TDD:

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 dependencies.

The following sequence of steps is generally followed:

1.Add a test – Write a test case that describe the function completely. In order to make the test cases the developer must understand the features and requirements using user stories and use cases.

2.Run all the test cases and make sure that the new test case fails.

3.Write the code that passes the test case

4.Run the test cases

5.Refactor code – This is done to remove duplication of code.

6.Repeat the above mentioned steps again and again

Benefits:

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

  Suitability:

- Well-suited for Agile development environments.
- Ideal for projects with clearly defined requirements and a focus on code quality and reliability.

BDD:

Behavioral Driven Development is a good approach in Automated Testing as it is more focused on the behavior of the system rather than the implementation of the code. BDD is facilitated through natural language to express the behavior of the system and the expected outcomes from the system. In BDD all parties are involved like a customer, developer, tester, and stakeholder for a collaborated conversation and illustration of the system's behavior.

1. Describe behavior –This includes the flow and features of the product means the main vision.

2. Define requirements –Modeled requirements with business rules for a shared understanding.
3. Run and fail the tests –Develop and run the test cases.
4. Apply code update –Refactor it according to the requirement.
5. Run and pass the tests –Run the updated code and pass the test cases.

Benefits: •   Greater clarity on business goals and customer

requirements.

- Reaches a larger customer set as it uses non-technical languages.
- Helps in defining acceptance criteria before development.
- Focuses on the system's behavior from the client's and developer's point of view.
- Helps in avoiding unnecessary features and includes important features.
- Reduces effort for post-modification and post-deployment defects.
- Avoids misinterpretations during the development process.
- Collaboration between team members is promoted through BDD, involving all stakeholders in the development process to ensure a shared understanding of objectives.

Suitability:

- Suitable for projects with complex business logic and a need for collaboration between technical and non-technical stakeholders.
- Ideal for Agile teams focused on delivering value to end-users.

FDD:

FDD stands for Feature-Driven Development. It is an agile iterative and incremental model that focuses on progressing the features of the developing software. The main motive of feature-driven development is to provide timely updated and working software to the client. In FDD, reporting and progress tracking is necessary at all levels. FDD Lifecycle

- Build overall model
- Build feature list
- Plan by feature
- Design by feature
- Build by feature Benefits:

- Reporting at all levels leads to easier progress tracking.
- FDD provides continuous success for larger size of teams and projects.
- Reduction in risks is observed as whole model and design is build in smaller segments.
- FDD provides greater accuracy in cost estimation of the project due to feature segmentation.

Suitability:

- Well-suited for large-scale projects with multiple development teams and diverse requirements.

Ideal for projects where features can be prioritized and developed independently