HARISH RONGALA
harishrongala123@gmail.com

Software Development Life Cycle and Agile Principles (part2)

Assignment 1: Create an info-graphic 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):

TDD, or Test-Driven Development, is not just for software only. It is also used to create product and service teams as test-driven work. To make testing successful, it needs to be created at both small and big levels in test-driven development. This means testing every part of the work, like methods in a class, input data values, log messages, and error codes. Other side of software, teams use quality control will check before starting work. These will be checks to help plan and check the outcomes of the tests.

They follow a similar process to TDD, with some small changes which are as follows:

1. “Add a check” instead of “Add a test”
2. “Run all checks” instead of “Run all tests”
3. “Do the work” instead of “Write some code”
4. “Run all checks” instead of “Run tests”
5. “Clean up the work” instead of “Refactor code”

Repeat these steps.

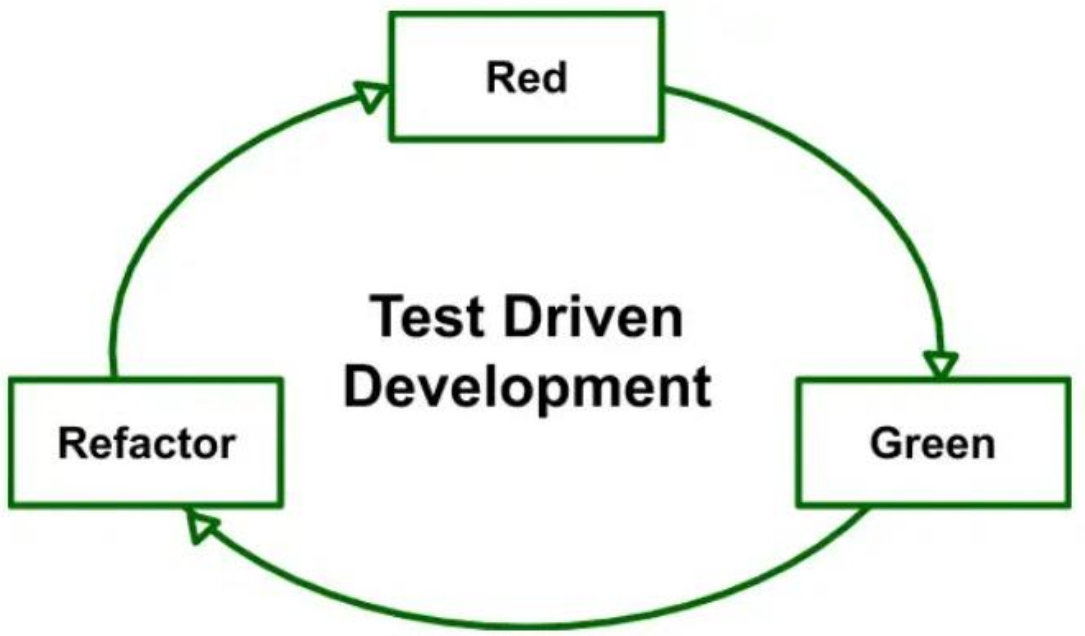
Process of Test Driven Development:

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

The following sequence of steps is generally followed:

1. Write a complete test case describing the function. 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



Red **:** Create a test case and make it fail
Green **:** Make the test case pass by any means.
Refactor **:** Change the code to remove duplicate/redundancy.

Advantages of Test Driven Development (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 life cycle.

Disadvantages of Test Driven Development (TDD)

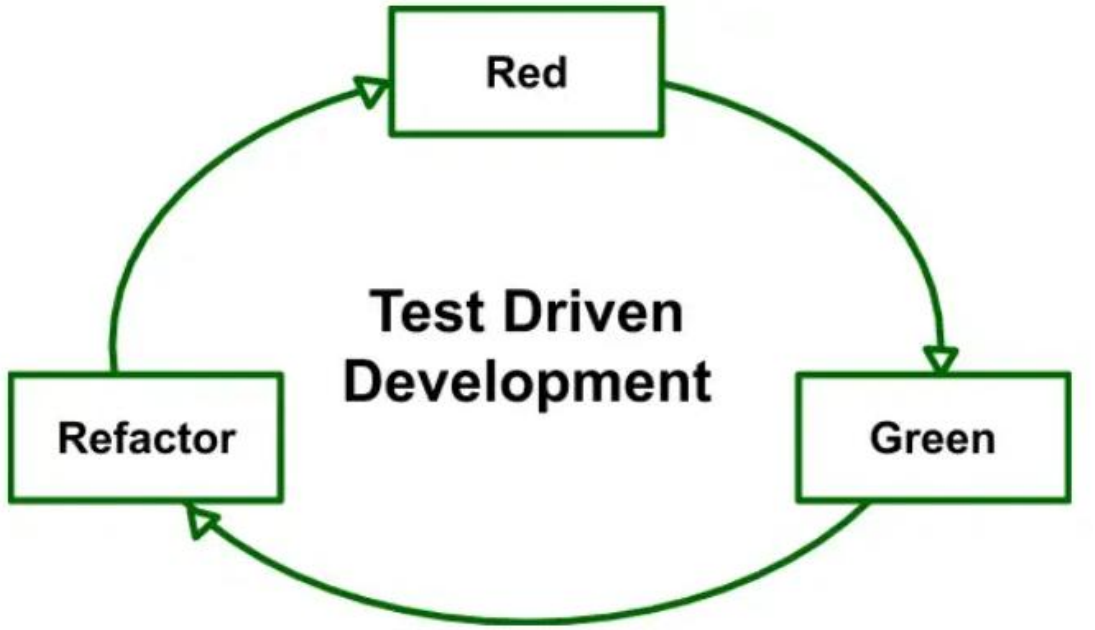
* Increased Code Volume: Using TDD means writing extra code for tests cases , which can make the overall code-base larger and more Unstructured.
* False Security from Tests: Passing tests will make the developers think the code is safer only for assuming purpose.
* Maintenance Overheads: Keeping a lot of tests up-to-date can be difficult to maintain the information and its also time-consuming process.
* Time-Consuming Test Processes: Writing and maintaining the tests can take a long time.

* Testing Environment Set-Up: TDD needs to be a proper testing environment in which it will make effort to set up and maintain the codes and data.

Assignment 2: Produce a comparative info-graphic 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) :

Test Driven Development (TDD) is a development technique which focuses more on the implementation of a feature of a software application/product. Mainly it refers to write a test case that fails because the specified functionality doesn't exist and after that update the code that can make the test case pass and as a result we get the feature implemented in the system. Actually it is a development practice where the developers are involved in it.



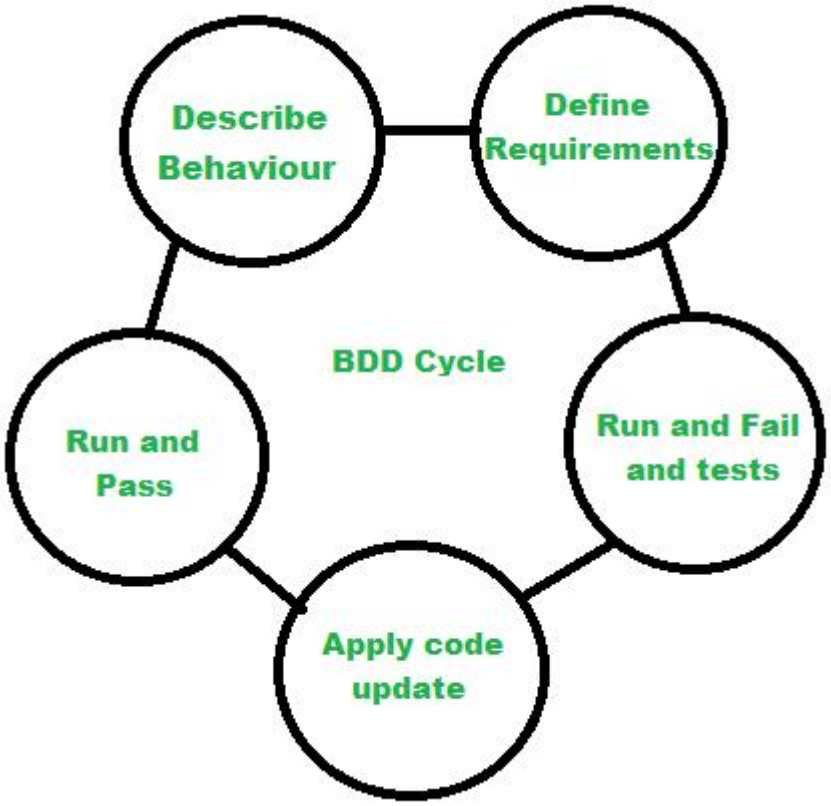
Red : Create a test case and make it fail
Green : Make the test case pass by any means.
Refactor : Change the code to remove duplicate/redundancy.

Process of TDD :

1) Add test case
2) Run the test cases and watch test fails
3) Update the code
4) Run the test cases again
5) Refactor the code (Optional)
6) Repeat the steps for another test case

Behavior Driven Development (BDD) :

Behavior Driven Development (BDD) is a development technique which focuses more on a software application's behavior. Mainly it creates an executable specification that fails because the respective feature doesn't exist, then writing the simplest code that can make the specification pass and as a result we get the required behavior implemented in the system. Actually it is a team methodology where Developers, Customer, QAs are involved in it.

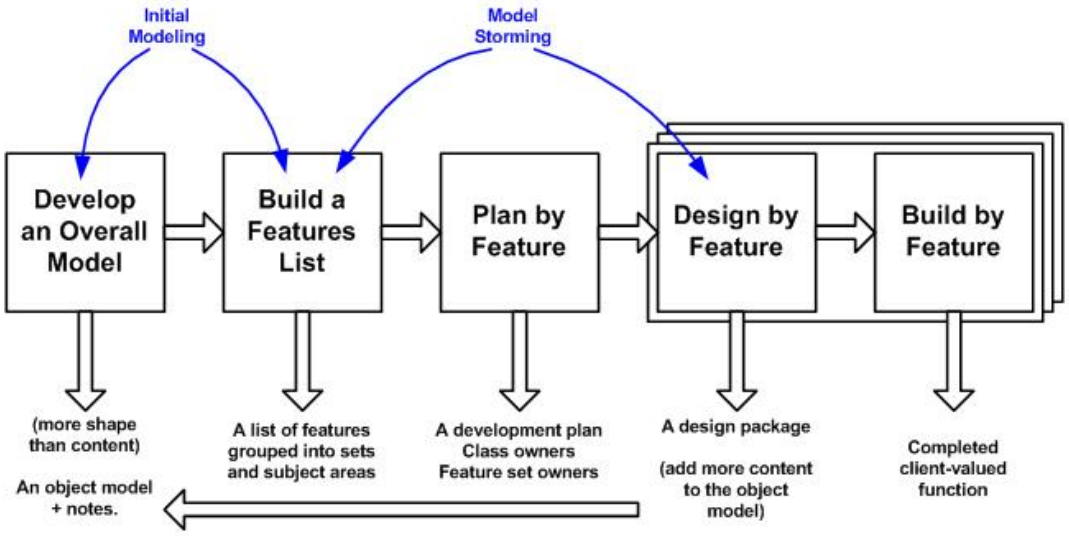


Process of BDD :

1) Write the behavior of the application
2) Write the automated scripts
3) Then Implement the functional code
4) Check if the behavior is successful and if not success then fix it
5) Organize the code (Optional)
6) Repeat the steps for another behavior

FDD(Feature-Driven Development):

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



Process of FDD:

1) Build overall model
2) Build feature list
3) Plan by feature
4) Design by feature
5) Build by feature