

Review of Test-Driven Development Strategy: Understanding Methods, Benefits and Limitations

Dr. Kamaldeep Garg

kamaldeep.garg@chitkara.edu.in

Assistant Professor

Chitkara University, India

Abstract: - Test driven development is the software development process which focuses on creating test cases before commencing the actual code writing. It is a process for development of systems and applications which uses iterative approach and integrates the process of programming, test cases etc. The test-driven development process implements variety of features of agile development and programming where testing process will guide the development of the software. This approach helps the testers and developers to obtain optimized code which is used for achieving long term goals. First of all, the testers will start creating test cases based upon the basic features of the application and then the code is written and started only when the test fails. This will help to stop the duplication of test scripts which in turn saves time and efforts. The paper will explain the various stages involved in following this approach and also explain the advantages and challenges of this development approach.

Keywords: - Introduction to test driven development, Characteristics of Test driven approach, Test driven development stages, Benefits of test-driven approach, Challenges of test-driven development.

Introduction: - [1]

Introductory long stretches of programming advancement include prerequisite get-together and examination by the business experts. They will sit with the client and assemble every one of the information and prerequisites from the client and afterward make a spending plan and really look at the funds and give proposition to the client. When the client supports, then, at that point, the essential report which depicts the prerequisites and particular of the task is given over to the product engineers who will begin coding the venture in light of the necessity detail archive. Whenever coding is finished, it is given over to the analysers to test the task and recognize the bugs. When the bugs are distinguished then designers will fix every one of them. This is reshaped till the venture is sans bug and is according to the client's necessities. Testing is the process which make sure that the application or program developed gives expected output when input is given. When talking about testing, it is classified into following two categories: -

1. Manual Testing: -

This type of testing is performed manually where the tester, once software is developed, uses it as a user and then make a list of all the bugs which he finds. Everything in this kind of approach is done manually. It is time consuming and requires a lot of effort. And in some scenario it is been observed that the manual testers are not able to identify all the bugs since it is done manually and as per human tendency they might miss out on something very important.

2. Automation testing: -

To overcome the disadvantages of manual testing, automation testing was discovered. This approach will write test scripts which executes automatically and identify all the bugs in the code written by the developer. The advantage of this type of approach is that it is much faster and efficient as compared to the manual testing approach. The other benefit is that the test scripts written can be reused in future testing as well.

Test-driven development: -

It is the latest innovation in the domain of software testing which is useful and is efficient as compared to the available testing approaches. In test driven development, the software is developed by first creating the test cases based upon the primary requirements of the software to be developed. Then the testers will start testing all the features by using these test scripts written. If they find out bugs then another iteration of this approach is executed till the time desired result is obtained. Once it is done then the actual code writing of the project starts. The biggest advantage of this type of approach is that it is efficient and give faster results when compared with other approaches. Test driven development has modified or one can say that has completely changed the traditional method of developing the software. In this approach, first a small test code is written which has minimal functional feature code. When this test fails, then another set of test script is written to pass the small functional feature. The whole process continues till the time the testers are satisfied with their test code. This approach verifies if the existing design is capable of implementing all the functions of the

project. If not then it will be refactored and then TDD is followed which is useful in making new designs which can be used in future. Test-driven development has following two levels: -

1. ATDD (Acceptance test driven development): -
 - It is also known as behaviour driven development where the aim is to identify detailed requirements which can be executed to provide solution which again based upon the just in time basis.
 - In this approach, acceptance test is conducted which will be enough to fulfil the test.
2. DTDD (Developer Test-driven development): -
 - With engineer TDD, compose a solitary designer test, here and there erroneously alluded to as a unit test, and afterward barely enough creation code to satisfy that test.
 - The objective of engineer TDD is to determine an itemized, executable plan for your answer on a JIT premise. Designer TDD is frequently just called TDD.

Characteristics of Test-driven development: - [2]

Following are features and characteristics of TDD: -

- It is the key factor which plays major role in test-driven development. The mindset of the team should be to provide and focus on verifying the function, validity as well as processing of the object by using efficient specific algorithms.
- This will help to remove the need for after code development testing requirements.
- Once the mindset if changes keeping in focus the business goals, the effectiveness of this approach is improved and provides many benefits.
- This makes sure that the interfaces used and classes defined are validated properly which will make sure that there is no duplication of the test cases or code.
- It is also mandatory to validate the design patterns.
- It is used to write simple and transparent code which helps the developer to maintain less documentation process.
- The inputs for TDD are acceptance test as well as the developer test cases.
- The objective of TDD is to promote testing of application code and detailed specifications.

Test Driven Development stages: - [3]

Following are the stages of the test-driven development approach: -



Figure 1 Test driven development stages.

1. Test case: -
 - In this first step, before writing the functional code, a test use case should be written to begin with for the component of the software which is to be developed.
 - The objective of this use case should be explained to the end user whether or not this use case passes or fails the test.
2. Writing unit test: -
 - In this step, a unit test will be created for the use case identified in the previous step. This is done before writing the implementation code for the use case.
 - Once it is done, then it is executed to see that it fails. The objective is that this test will surely fail and if it does not fail, then there must be something wrong in initial stages only.
 - This could be indication of using a duplicate test case or using the same variable which is already in use.
3. Check of the test case passes: -
 - In this step, only that much code is written which is sufficient to pass the test.
 - Once this test is pass, then one is good to go for writing another test case for the next use case.
4. Compose a test for the following use case in a similar way as the primary experiment. Along these lines, we are developing a test suite that develops naturally with the product to cover practically all executable code.
5. In this step, again test cases are written which will pass all the existing cases as well as the new test case. This process is repeated until all the test cases have been

written for the corresponding feature of the application being developed.

6. Refactor: - In this step, the code will be refactored until the base of the code is cleaned up properly.
7. It is iterative process, so the steps 4-6 are repeated till the time one gets complete code written based upon the requirements.

Code colour of the Test-driven development: - [4]

The stages of the TDD methodology can be understood by following code colour of various stages of the TDD: -

1. Code Red: -
 - This implies the failure of the test case. The first test will fail as there will be no code at all in initial stages to be compiled.
 - This stage is known as red stage code which means failure of the testing of the code.
2. Green Code: -
 - In this stage, a code is written and tested to have pass as result.
 - This is so as the developer will write only small piece of code which will be useful to pass the test which has failed in the previous step.
 - This is challenging for the developers because they are use to of writing code for the complete module of the project.
3. Green code: -
 - This code describes and makes sure that all the existing test cases passes along with each new test case written.
 - It is a kind of regression testing which makes sure that along with every new written test case, all the old test cases also passe.
4. Refactoring: -
 - The aim of this stage is to make sure that the functionality of the module is same and does not give error and refined.
 - Refactoring means cleaning the code to make sure that the designs, and code can be maintained efficiently.
5. Repeat steps 1-4: -

All the steps are repeated in the TDD cycle until all the features are implemented.

Advantages of Test-Driven Development: - [5]

Following are the benefits of test-driven development: -

1. Easy maintenance: -

Since there are transparent interfaces, and various parts of the system is decoupled from each other and does not depend upon each other, the code is easy to maintain. Even the implementation of one module can be done without interfering with other modules.
2. Modular Designs: -

In TDD approach, since one function of the application is considered one at a time due to which it is easy to test the code due to its modular design.
3. Easy to refactor: -

In this approach, since all the features are implemented and tested in details one after the other so there will be no big modification to be made and will be easy to refactor the code whenever required.
4. Less debugging time: -

Since each feature is tested in detail, there will be less debugging time and the project is delivered on time.
5. Provides better and detailed coverage area: - The test-driven development process allows for providing better and high testing coverage area as a result of which the application developed using this approach is efficient as compared to traditional methods of development.

Disadvantages of Test-driven Development: -

Other than benefits of test-driven approach, there are few limitations also of the technique which are discussed below:

1. Slow development process: - Since it provides efficient quality application but it is time consuming process as the time taken to write test cases for each feature before the actual coding of the functions is more.
2. Difficult to make modifications when requirement changes: - Whenever the requirements are changed, it is difficult to make changes in the code as the whole process is to be repeated to accommodate this change.
3. TDD should be used by all in the team: - When it is decided that TDD will be used for developing the application, then it has to be used by each and every member of the team.

Conclusion: -

Test driven improvement is the product advancement process which centres around making experiments prior to initiating the real code composing. It is an interaction for improvement of frameworks and applications which utilizes iterative methodology and incorporates the most common way of programming, experiments and so on. The test-driven improvement process executes assortment of highlights of spy turn of events and programming where testing cycle will direct the advancement of the software. This approach helps the analysers and designers to acquire upgraded code which

is utilized for accomplishing long haul objectives. Above all else, the analysers will begin making experiments in light of the essential elements of the application and afterward the code is composed and begun just when the test fizzles. This will assist with halting the duplication of test scripts which thus saves time and endeavours. There are limitations to this approach. For improving the test-driven approach, behaviour driven development is discovered which removes the short comings of the test driven approach.

Reference: -

- [1]. <https://www.freecodecamp.org/news/an-introduction-to-test-driven-development-c4de6dce5c/>
- [2]. <https://triare.net/insights/test-driven-development/>
- [3]. <https://www.andplus.com/blog/test-driven-development>
- [4]. <https://www.whizlabs.com/blog/what-is-tdd-and-its-phases/>
- [5]. <https://www.geeksforgeeks.org/advantages-and-disadvantages-of-test-driven-development-tdd/>