A Study of Framework of Behavioural Driven Development: Methodologies, Advantages, and Challenges

Dr. Rupali Gill
Professor, Chitkara University, India
rupali.gill@chitkara.edu.in

Abstract: - BDD is the behavioural driven development which is used for the development process of the software. It used to describe behaviour of the system in a language which is easy to read and understand. It is the development process which is used to bridge the gap between the technical people involved in the project and the businesspeople who are not technically sound. The aim and goal of BDD is to deliver business value products for which it helps to provide communication between developers and analysts via shared process and tools. The origin of BDD is TDD which is test driven development where examples are used to describe the behaviour of the system which could be used as acceptance test, and which could be converted into executable specifications. This process is very useful for the software developers as it is very useful for them to develop a software project which fulfils all its goals and aim. This methodology also helps to get rid of the confusion and improves the communication process between the developers, testers, and product managers. This is derived by making few modifications in the already existing test-driven development and now BDD is considered an appropriate methodology which is an enhanced software development process. The paper will explain the importance of BDD, phases of BDD, application use etc.

Keywords: - Introduction to Behavioural driven development, Importance of BDD, Life cycle of BDD, Application use, Advantages, Limitations.

Introduction: - [1]
Initial days of software development involves requirement gathering and analysis by the business analysts. They will sit with the client and gathers all the data and requirements from the client and then make a budget and check the finances and give proposal to the client. Once the client approves, then the necessary document which describes the requirements and specification of the project is handed over to the software developers who will start coding the project based upon the requirement specification document. Once coding is done, it is handed over to the testers to test the project and identify the bugs. Once the bugs are identified then developers will fix all of them. This is repeated till the project is bug free and is as per the client’s requirements. Now-a-days, business need methodologies which involves efficient working capabilities, and which can complete the project at a much faster pace so the developers should also meet all the business demands. To meet these objectives and to make sure that the quality of the developed software is not compromised, businesses have started using agile methodologies. One of the latest agile methodologies is BDD which is behaviour driven development. It is the method of developing software which helps to describe the behaviour of the system with the help of examples and helps to get rid of the communication gap between various team members involved in the software development. It helps to collaborate the developers, testers, businesspeople, etc. It describes the behaviour of the system in simple language with the help of real-world examples which is easy to understand technical as well as non-technical people. This latest methodology has emerged from test driven development. In TDD, test scripts are written before coding of the software is done and to do coding all the tests should pass. BDD uses most of the principles and methods of TDD and helps to collaborate the teams with relevant project information.

Importance of BDD: - [2]
BDD is derived from TDD so it is important for the following reasons: -

➢ Since BDD uses the concept of TDD, so the tests written will focus more on how the system behaves under different scenarios.
➢ Various elements of the system will be developed based on the behavioural predictions.
➢ It is efficient in terms of cost of the testing of the project.
➢ It also saves time and effort which is used to identify and fix the shortcomings after the deployment is done.
➢ It uses simple non-technical language to describe the goals and objectives.
➢ It also put more focus on the way an application behaves as per the end user.
➢ All the situations are written in shared language which is easily understood by all the team members involved in the development of the software.
➢ It has the most advanced and latest tools like cucumber, Mspec, SpecFlow etc.
➢ BDD uses automated scripts which is used to convert the behaviour predictions into tests.

**Behavioural Driven Development process/Implementation:**

BDD implementation process is a three-step process which is summarised as following:

1. **Discover:**
   - The goal of the business is to deliver a software which has high quality. One way to achieve this is by having communication between all the people involved in the software development phase.
   - The first step is to have user story which is considered as upcoming change to the system and should discuss the new functionality.
   - Before starting the process, the team can use various email marketing tools to understand the user’s perspective and gather user’s ideas.
   - By doing so, everybody involved in the team will be able to understand the details of the goals to be achieved.
   - After this test scripts are written in a language which is easily understood by everybody involved in the process and it will give description about: - a. How to test the application, b. The actual test, c. the behaviour of the application.
   - Real time examples are used to write test cases to explain the actual requirements which will further explains how the system behaves.
   - Real time examples which explain the user’s needs are used as structured conversations which are easy to understand by technical as well as non-technical people in the team involved for the software development of the project.
   - It also makes sure that it explains the scope of the project which explains what is to be done, and how the system must function.

2. **Formulation:**
   - This explain that how the process is formed to develop the software, and which can be understood easily by everybody in the team.
   - It explains that how what it should do and in this step all the documented examples are converted into automated scripts after which it is tested for agreement.
   - Once from the first stage, valuable example is identified, then it will be converted into structured documentation.
   - By doing so, everybody in the team can be asked for their agreement that they understand the shared understanding of
   - BDD uses documentation process which is easy to understand by all the people and systems and aim is to make sure:
     a. To automate examples which can provide guidance to the team members to develop project.
     b. To have a shared vision of the team by taking feedback from each team member involved in the process.

3. **Automation:**
   - This is the next step once formulation is done which explains what it actually does in reality.
   - This is the final step which is done in order to implement each behaviour example and to begin with automated test will be written which will guide the code development process.

---

**Figure 1: Behavioural driven development implementation process.**
Once the team has access to the executable form of the specification, it can be used for the implementation process.

After this, each example will be connected to the system as a test which will fail as the behaviour which explains this has not been implemented yet.

In the next step, the implementation code is developed. Behaviour of the internal components of the system are used as a guide.

The feedback taken at every stage helps to remove the effort of performing test manually which in turn helps people to spend time and effort in other product tasks.

Making each modification minor and iterating quickly are the goals of this three-step procedure. Each time more data is needed, you can advance a level. Every time a fresh example is automated and put into practise, it signifies you’ve improved the system and are now prepared to react to criticism.

Application use or best practices for BDD: -
Following are few best practices for Behavioural driven development: -

1. Short Background: -
   - The background should always be kept short which should be same for each scenario in future files.
   - The proper way of using the background allows to avoid duplicate files in future.
   - Background should use generic interface.

2. Featured files: -
   - Single format should be selected for features of the system.
   - One must avoid detailed descriptions.
   - All the featured files must contain same format as selected.
   - This helps newcomers to understand what is going on in the project.

3. Steps and scenarios: -
   - Implementation details should be hidden and not visible.
   - In order to keep the scenarios short, technical details should also be hidden.
   - Long, detailed scenarios with numerous “AND” or “BUT” phases become fragile and difficult to read.
   - Therefore, even though you shouldn’t include implementation specifics in your steps, there should be enough information for the scenario's context to be understood.

4. Tags: -
   - Relevant tags name should be used which can be understood by its name itself, so it should have meaningful name.
   - Tag the features smartly.
   - Tags should not be used for unimportant scenarios. This means if a tag is used for one scenario may not be used for another.

Advantages of BDD: - [4]
Following are some advantages of BDD: -

- Efficient collaboration: - In BDD simple language is used which is easy to understand, this helps in collaboration between developers, testers, technical staff as well as non-technical staff.
- Good Visibility: - Since the language used explains the behaviour of the system to be developed it has high visibility as compared to traditional methods.
- More focus on User requirements: - BDD methodology helps in delivering the product on time and focuses on user needs which results in user satisfaction.
- Aims and goals of business are met: - All the business aim and objectives of the business are met using BDD methodology.

Disadvantages of BDD: -
Following are few challenges of BDD approach: -

- Working knowledge of TDD ideas is necessary for BDD.
- The waterfall approach and behavior-driven development are incompatible.
- BDD testers must possess the necessary technical knowledge.
- If the needs are not carefully articulated, BDD may not be successful.
Conclusion: BDD is the conduct driven improvement which is utilized for the advancement cycle of the product. It used to depict conduct of the framework in a language which is not difficult to peruse and comprehend. It is the improvement cycle which is utilized to overcome any barrier between the specialized individuals associated with the undertaking and the finance managers who are not in fact sound. The point and objective of BDD is to convey business esteem items for which it assists with giving correspondence among designers and examiners by means of shared interaction and apparatuses. The beginning of BDD is TDD which is test driven improvement where models are utilized to depict the way of behaving of the framework which could be utilized as acknowledgment test, and which could be changed over into executable details. This cycle is extremely valuable for the product engineers as it is exceptionally helpful for them to foster a product project which satisfies every one of its objectives and point. This procedure likewise assists with disposing of the disarray and further develops the correspondence cycle between the designers, analyzers, and item chiefs. This is determined by making not many alterations in the generally existing test-driven improvement and presently BDD is viewed as a fitting procedure which is an upgraded programming advancement process.

References:

[1]. https://www.leapwork.com/blog/introduction-to-behavior-driven-development


[3]. https://www.cuelogic.com/blog/bdd-vs-tdd

[4]. https://www.altexsoft.com/blog/behavior-driven-development/#:~:text=


[6]. https://www.altexsoft.com/blog/behavior-driven-development/