**TDD ADVANTAGES:**

1. Better program design and higher code quality
2. Detailed project documentation
3. TDD reduces the time required for project development
4. Code flexibility and easier maintenance
5. With TDD you will get a reliable solution
6. Save project costs in the long run

**TDD DRAWBACKS:**

* Forget to run tests frequently.
* Write too many tests at once.
* Write tests that are too large.
* Write tests that are overly trivial.
* Write tests for trivial code.

**TDD AND BDD DIFFERENCES:**

|  |  |
| --- | --- |
| TDD | BDD |
| Stands for Test Driven Development | Stands for Behavior Driven Development. |
| The process starts by writing a test case. | The process starts by writing a scenario as per the expected behavior. |
| TDD focuses on how the functionality is implemented. | BDD focuses on the behavior of an application for the end user. |
| Test cases are written in a programming language. | Scenarios are more readable when compared to TDD as they are written in simple English format. |
| Collaboration is required only between the developers. | Collaboration is required between all the stakeholders |
| Some of the tools which support TDD are: JUnit, TestNG, NUnit, etc. | Some of the tools which support BDD are SpecFlow, Cucumber, MSpec, etc. |
| Tests in TDD can only be understood by people with programming knowledge. | Tests in BDD can be understood by any person including the ones without any programming knowledge. |
| TDD reduces the likelihood of having bugs in your tests. | Bugs in tests are difficult to track when compared to TDD. |
| Changes in how the application functions impact a lot on the test cases in TDD. | BDD scenarios are not much impacted by the functionality changes. |