

When comparing Test Driven Development and Behavior Driven Development, it's important to know the definition of each. Test Driven Development, or TDD, is when tests are designed first and code is written to pass the tests. Usually the tests are evaluating the functionality of small parts of code. The programmer refactors their code so that it will function as expected and pass the test. Where as Behavior Driven Development, or BDD, is more focused on what the user will experience and code is written based off that. BDD also tends to incorporate more team input, including members from different departments.

<https://www.guru99.com/test-driven-development.html>

<https://www.techtarget.com/searchsoftwarequality/definition/Behavior-driven-development-BDD>

<https://www.softwaretestinghelp.com/tdd-vs-bdd/>

<https://www.tutorialspoint.com/state-the-differences-between-tdd-and-bdd>

Mocking allows a developer to create fake dependencies, which are based on the actual dependencies in the code, to be able to isolate and test a specific class with more control. Mocking also allows a developer to examine how the code that is being tested interacted with it's dependencies. A very popular open source framework for mocking in Java is Mockito.

<https://semaphoreci.com/community/tutorials/stubbing-and-mocking-with-mockito-2-and-junit>

<https://www.vogella.com/tutorials/Mockito/article.html>