Behavior Driven Development

Behavior Driven Development (**BDD**) is a software engineering process that stems from Test Driven Development (TDD) and Acceptance Test Driven Development (ATDD).

 Behavior-driven development combines the general techniques and principles of TDD with ideas from [domain-driven design](https://en.wikipedia.org/wiki/Domain-driven_design) and [object-oriented analysis and design](https://en.wikipedia.org/wiki/Object-oriented_analysis_and_design) to provide software development and management teams with shared tools and a shared process to collaborate on software development

**[Domain-driven design](https://en.wikipedia.org/wiki/Domain-driven_design)**[(](https://en.wikipedia.org/wiki/Domain-driven_design)**[DDD](https://en.wikipedia.org/wiki/Domain-driven_design)**[) is the concept that the structure and language of software code (class names, class methods, class variables) should match the](https://en.wikipedia.org/wiki/Domain-driven_design)[business domain](https://en.wikipedia.org/wiki/Business_domain" \o ")[.](https://en.wikipedia.org/wiki/Domain-driven_design)

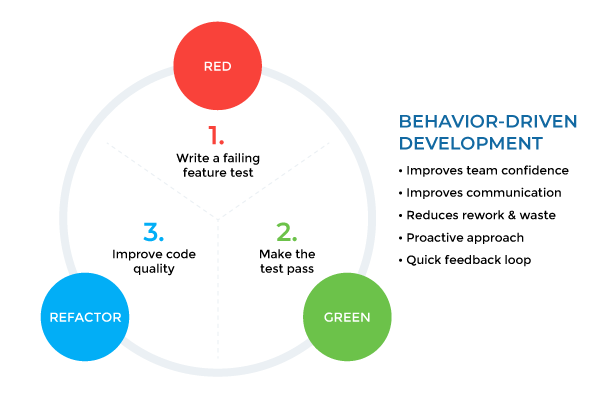
[Behavior-driven development is an extension of](https://en.wikipedia.org/wiki/Domain-driven_design)[[test-driven development](https://en.wikipedia.org/wiki/Domain-driven_design)](https://en.wikipedia.org/wiki/Test-driven_development)[:](https://en.wikipedia.org/wiki/Domain-driven_design)[[[9]](https://en.wikipedia.org/wiki/Domain-driven_design)](https://en.wikipedia.org/wiki/Behavior-driven_development#cite_note-ATDD_vs_BDD-9)[development that makes use of a simple, domain-specific scripting language (DSL)](https://en.wikipedia.org/wiki/Domain-driven_design)

BDD focuses on:

* Where to start in the process
* What to test and what not to test
* How much to test in one go
* What to call the tests
* How to understand why a test fails

Principles of BDD

* define a test set for the unit *first*;
* make the tests fail;
* then implement the unit;
* finally verify that the implementation of the unit makes the tests succeed.



**TDD vs. BDD**

* BDD is simply written in plain English language that is easy to understand by all members of the team while TDD is written in complicated IT languages like Ruby on Rails, Java, Python etc.
* BDD demonstrates the application behavior keeping in view the potential users, in contrast, TDD only focuses on functional implementation. This is how quality and CD can be more efficiently maintained in BDD as compared to TDD.

