Behavior Driven Development

**In** [**software engineering**](http://en.wikipedia.org/wiki/Software_engineering)**, behavior-driven development (abbreviated BDD) is a** [**software development process**](http://en.wikipedia.org/wiki/Software_development_process) **based on** [**test-driven development (TDD)**](http://en.wikipedia.org/wiki/Test-driven_development)**.**[**[1] HYPERLINK "http://en.wikipedia.org/wiki/Behavior-driven\_development"[2]**](http://en.wikipedia.org/wiki/Behavior-driven_development) **Behavior-driven development combines the general techniques and principles of TDD with ideas from** [**domain-driven design**](http://en.wikipedia.org/wiki/Domain-driven_design) **and** [**object-oriented analysis and design**](http://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**.[[1 HYPERLINK "http://en.wikipedia.org/wiki/Behavior-driven\_development"] HYPERLINK "http://en.wikipedia.org/wiki/Behavior-driven\_development"[3]](http://en.wikipedia.org/wiki/Behavior-driven_development)

Although BDD is principally an idea about how software development should be managed by both business interests and technical insight, the practice of BDD does assume the use of specialized software tools to support the development process.[[2]](http://en.wikipedia.org/wiki/Behavior-driven_development) Although these tools are often developed specifically for use in BDD projects, they can be seen as specialized forms of the tooling that supports test-driven development. **The tools serve to add automation to the** [**ubiquitous language**](http://en.wikipedia.org/wiki/Domain_driven_design) **that is a central theme of BDD.**

* **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**

**At the heart of BDD is a rethinking of the approach to the** [**unit testing**](http://en.wikipedia.org/wiki/Unit_testing) **and** [**acceptance testing**](http://en.wikipedia.org/wiki/Acceptance_testing) **that came up with while dealing with these issues.** For example, proposing that **unit test** names be whole sentences starting with the word **"should**" and **should be written in order of business value i.e., essentially states that for each unit of software. Acceptance tests should be written using the standard agile framework of a** [**User story**](http://en.wikipedia.org/wiki/User_story) **ie., which is each explicitly written down in a dedicated document: "As a [role] I want [feature] so that [benefit]". Acceptance criteria should be written in terms of scenarios and implemented as classes: Given [initial context], when [event occurs], then [ensure some outcomes].**[**[1]**](http://en.wikipedia.org/wiki/Behavior-driven_development)

**BDD framework developed as a communication and collaboration framework for developers,** [**QA**](http://en.wikipedia.org/wiki/Quality_assurance) **and non-technical or business participants in a software project.**

BDD is a second-generation, outside-in, pull-based, multiple-stakeholder, multiple-scale, high-automation, agile methodology. It describes a cycle of interactions with well-defined outputs, resulting in the delivery of working, tested software that matters.

**Principles of BDD**

**At its core, behavior-driven development is a specialized version of test-driven development which focuses on behavioral specification of software units**.

**Test-driven development is a software development methodology which essentially states that for each unit of software, a software developer must:**

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

Behavioural specifications

**Following this fundamental choice, a second choice made by BDD relates to *how* the users' desired behavior should be specified. In this area BDD chooses to use a semi-formal format for behavioral specification which is borrowed from** [**user story**](http://en.wikipedia.org/wiki/User_story) **specifications from the field of** [**object-oriented analysis and design**](http://en.wikipedia.org/wiki/Object-oriented_analysis_and_design)**. BDD specifies that business analysts and developers should collaborate in this area and should specify behavior in terms of user stories, which are each explicitly written down in a dedicated document.**

**Specialized tooling support**

Much like test-driven design practice, behavior-driven development assumes the use of specialized support tooling in a project. In as much as BDD is, in many respects, a more specific version of TDD, the tooling for BDD is similar to that for TDD, but **makes more demands on the developer than basic TDD tooling**

<http://en.wikipedia.org/wiki/Behavior-driven_development>