Behavior-Driven Development (BDD) - Basics

Duration: 1 day
Audience: Chargé de projet

Who is this training for?

- Functional analysts
- Developers
- Functional architects and software architects.

Training objectives

- Describe the BDD, what it consists of and why to use it
- Define the functional requirements with specifications and examples
- Organize everything with user stories and iterative development
- Improve communication in order to maximize the business value of the products delivered by the development team

Summary

Imagine not only being able to automate your unit tests, but also functional tests. To be able to understand a need only by reading a few examples. Bridge the communication gap between business and development. In short, go from specifications to examples, then to design the code while always having the vision and business objectives in mind. The BDD or Behavior-Driven Development approach was invented, then perfected, to be able to respond to all this. This introduction teaches the fundamentals of this approach and everything that is needed to implement it effectively through an agile development team.

Course outline

Definition of BDD

Copyright © 2026 AFI by Edgenda inc.

- What is the problem?
- Why and when to use it?
- The place among other types of tests The BDD/TDD cycle

Vision and impacts

- Why do we do this?
- The "Impact Mapping" approach
- Planning our iterations and deliveries through "story mapping"

The "Gherkin" or "Given-When-Then" formalism

- Description
- How to use it, even in French!

Specification workshop

- Exploring, discovering and defining behaviors
- How do we organize it?
- The Importance of the "3 Amigos"

Common Problems

- Tips and Tricks
- Patterns of Specification by Example
- Test Scripts vs. Specifications
- Starting with Bad User Stories

Tools

- Brief Overview of the Possibilities Tools Can Bring
- Reports and Automation

Approach and methodology

Copyright © 2026 AFI by Edgenda inc.

Lecture presentation with several activities and reflections to be done alone or in teams. A fictitious case will be used throughout the day and will evolve according to the concepts learned. Please note that there will be no technical exercises, i.e. programming, during this training. That's why it is aimed at both the analysis and development side.

Prerequisites

Have a basic knowledge of agility.

Recommendations

Familiarize yourself with the basic concepts: agile methods (Scrum, user stories) software development cycle Identify a real project context (requirements, business-IT communication) Think about your current challenges: misunderstanding of needs, communication defects between teams, quality of specifications.

Copyright © 2026 AFI by Edgenda inc.