ACAMICA

Globant
we are ready

# Level 6

# Introduction to BDD

# 6.1 - Introduction

**Behavior-driven development** (BDD) is a software process resulting from the process known as **Test-driven development** (TDD).

**BDD** conceives software development as the conjunction of technical and business aspects and interests.

**BDD** seeks to unify the understanding of the actors involved in software development through the definition of a domain-specific language (DSL) that uses English words and logical constructs to express the behavior of the software to be developed.

**Given - When - Then** is a scenario writing style, which conforms to BDD’s description to provide a language that is shared by all the members of a software development team, including software developers and non-technical personnel, that helps describe software behavior clearly and concisely.

The essential idea is to create a scenario based on the following sections:

- **Given**. The state prior to the beginning of the behavior to be described is described, we can say that it is a pre-condition of our scenario.
- **When**. The behavior is described.
- **Then**. The changes that we expect from the behavior described above are specified.

Next we will see a scenario specified with **Given** - **When** - **Then**.

```
Feature: User trades stocks
  Scenario: User requests a sell before close of trading
    Given I have 100 shares of MSFT stock
      And I have 150 shares of APPL stock
      And the time is before close of trading

    When I ask to sell 20 shares of MSFT stock

     Then I should have 80 shares of MSFT stock
      And I should have 150 shares of APPL stock
      And a sell order for 20 shares of MSFT stock should have been executed
```

# 6.2 - Gherkin

The scenario we saw earlier is written in a language called **Gherkin**.

This language allows us to describe the functionalities, defining software behavior, without entering into software implementation.

Through **Gherkin**, the behavior of the software is documented in terms of scenarios, and these scenarios can be executed by automated test cases, from their description.

**Note:** Gherkin supports over 60 languages.

Let’s dive into the description of the sections that make up a scenario written in Gherkin.

- **Feature**. Name of the functionality that we are going to test. It must be clear and concise.
- **Scenario**. It describes the scenario we are going to test.
- **Given**. It provides the initial context for the scenario that is going to be tested.
- **When**. It specifies the set of actions to be executed on our application under test.
- **Then**. The expected state from the executions is in the **When** section.

# 6.3 - BDD Frameworks

Next we are going to list some of the available BDD frameworks for JVM languages.

## JBehave

**JBehave** is a BDD framework developed in Java that allows the development and execution of user stories written in the **given-when-then** format.

**JBehave** can be easily integrated with JUnit and TestNG. It also incorporates a module through which it is possible to work with WebDriver.

We can use **JBehave** with Groovy, Java, Scala and JRuby.

## Cucumber JVM

**Cucumber JVM** is the implementation of the Cucumber framework for languages that work with JVM, such as: Java, Groovy, Scala, Gosu, Jython, Rhino JS, etc.

**Cucumber JVM** provides several mechanisms to execute user stories written using the Gerkin language. Within these mechanisms, we can use: JUnit, CLI Runner (useful to execute stories on a continuous integration server) and different plugins available, for example for Eclipse IDE.

## Concordion

**Concordion** proposes that the writing of user stories be made through HTML files. These same HTML files act as documentation of the software and, in turn, are executable specifications (through JUnit), since Concordion is responsible for relating specifications with Java code.

**Concordion** also supports that user stories are made through Excel documents, it has integration with Jenkins and provides an API to create customized extensions.

# Summary of the Level

At this level we have seen a definition of the **Behavior-driven development** (BDD) process, we have mentioned its characteristics and have dived into the **Given - When - Then** scenario writing style.

We have introduced the **Gherkin** language, and finally we mentioned the most popular frameworks, available for JVM, to work with BDD.