BDD – Writing Test Scenarios

21TH JAN, 2025

expleo

Contents

Introduction

- Behaviour-driven development, also called BDD, is a methodology that involves developing software through examples and provides collaboration between different stakeholders on the team.
- Stakeholders on a SDLC team are a mixture of technical and non-technical people.
- BDD encourages all team members to work closely using a common, non-technical language to set the expectations and behavior of a feature.
- Because of BDD, they share a common understanding of the work that they have to deliver.
- Collaboration and communication clear any ambiguity, misunderstandings, and differences in opinions among the team.

BDD - Benefits

- Better communication and collaboration;
- Shared understanding;
- Less ambiguity;
- Executable artifacts;
- Living Documentation.

Using Scenarios – BDD is designed to speed up the development process. Everyone involved in development relies upon the same scenarios. Scenarios are requirements, acceptance criteria, test cases, and test scripts all in one; there is no need to write any other artifact.

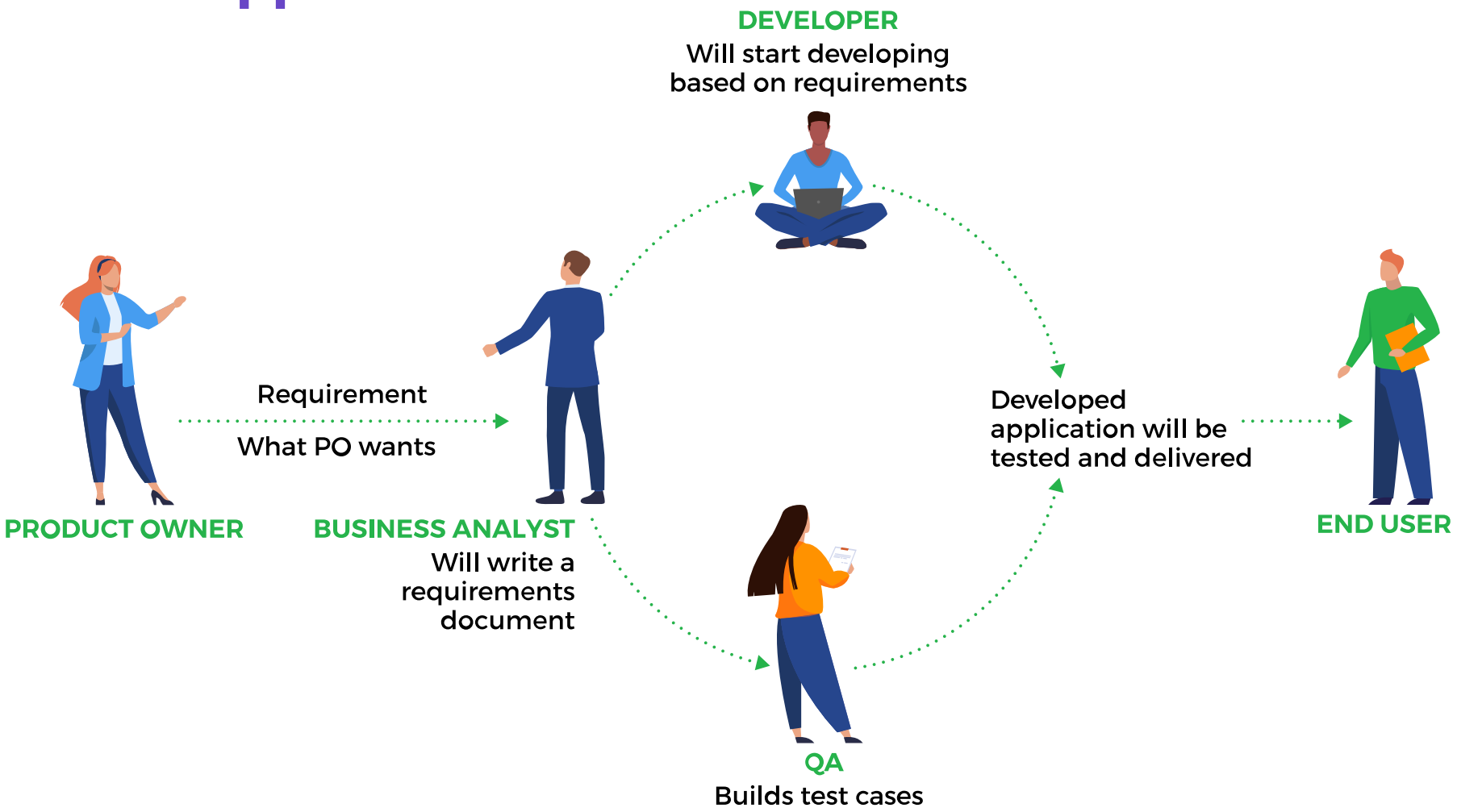
Principles of BDD

- BDD encourages simple languages to be used across teams, known as ubiquitous languages.
- The simple and easy to use language should be used in the way the tests themselves are written, so that in theory, a business person can read a test and understand what it is testing.
- Tests are often written from the customer's point of view; the focus is on the customers and the users who are interacting with the product

Traditional Approach

- In a traditional SDLC without BDD, the business analyst, developer and tester work independently.
- This process leaves a high chance of miscommunication around understanding the requirements.
- The developer will have one understanding and based on that, the application will be developed. While the tester will design test cases based on their own understanding.

Traditional Approach



expleo

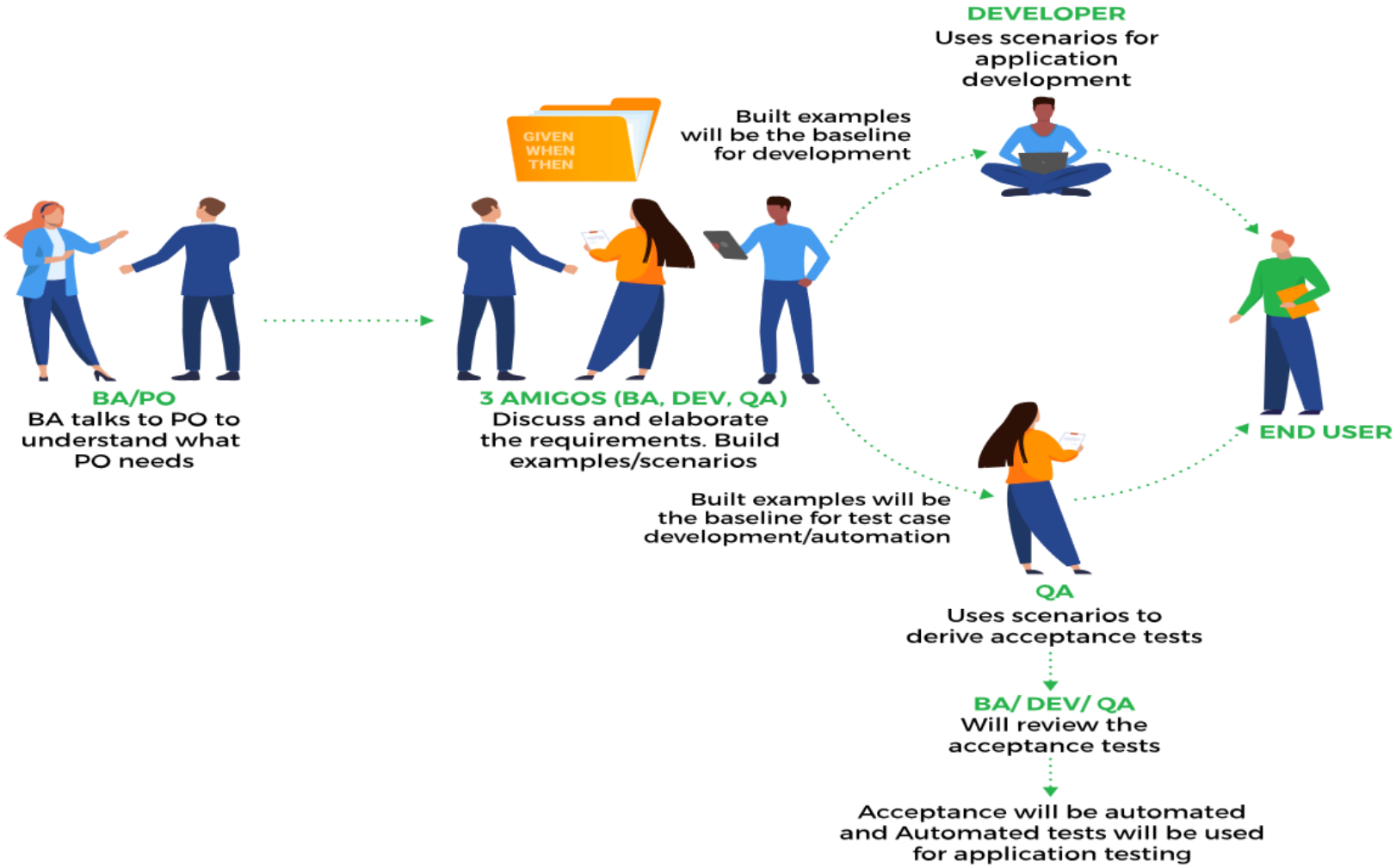
BDD Approach

➢ The three amigos are the business analyst, developer, and tester. At the start of a project, they meet to discuss and elaborate on the requirements using examples.

- They speak through examples;
- Discuss how the features will behave in different situations;
- Build rules around the expectations;
- Come up with a set of working examples for the features.

Through this process, all stakeholders understand the requirements and how they should behave in various situations. The examples will become scenarios in BDD.

BDD Approach



expleo

Gherkin - DSL

- Once every stakeholder has a shared understanding, they must document every scenario that they agreed on. The structured way of writing them is known as Gherkin Language. Gherkin provides a set of keywords, they are:

  Given - is a precondition or current state of the system;

  When - is the action which happens on the system;

  Then - is the outcome of the event that happened on the 'When' step.

- These keywords are also often referred to as Domain Specific Language (DSL). With BDD Domain Specific Language, the team writes down the scenarios using the language that is known to them in a structured format and defined terminologies.

- This way, everyone is clear about what they want to achieve, and there is no room for confusion.

Gherkin - DSL

- In addition to that, you can use And after any of the descriptors if you require more information in BDD scenarios.

Given (context),

And (some more context),

When (action/event),

And (further action/event),

Then (outcome)

And (further outcome)

Mapping User Stories using BDD – Example 1

- **Let's look at an example of BDD in action. Assume that the new feature is going to be an "Add to Cart" functionality of an e-commerce application.**

- The three amigos (business analyst, developer, and tester) discuss what that feature is and how the feature will behave in different scenarios.
- They will take different examples and see how the feature will behave in each case.
- These examples will be the scenarios for developing the feature.
- Once they agree on each case, they will "formally" write down different scenarios - which are the examples.

Mapping User Stories using BDD

- First, they will decide on pre-requisites/pre-conditions/current systems. For example: to implement the 'Add to Cart" feature, they come up with the following prerequisites:

  Customer searched for a product in the application

- So that goes to the Given statement of the example:

  Given a customer has searched for a product

Mapping User Stories using BDD

- Now that they have to decide on actions associated with the scenario-

  Customer adds product to the cart

- It can be written using When statement as below:

  When customer adds product to the cart

Mapping User Stories using BDD

- Finally, the team identifies the outcome of the action and described them with Then:

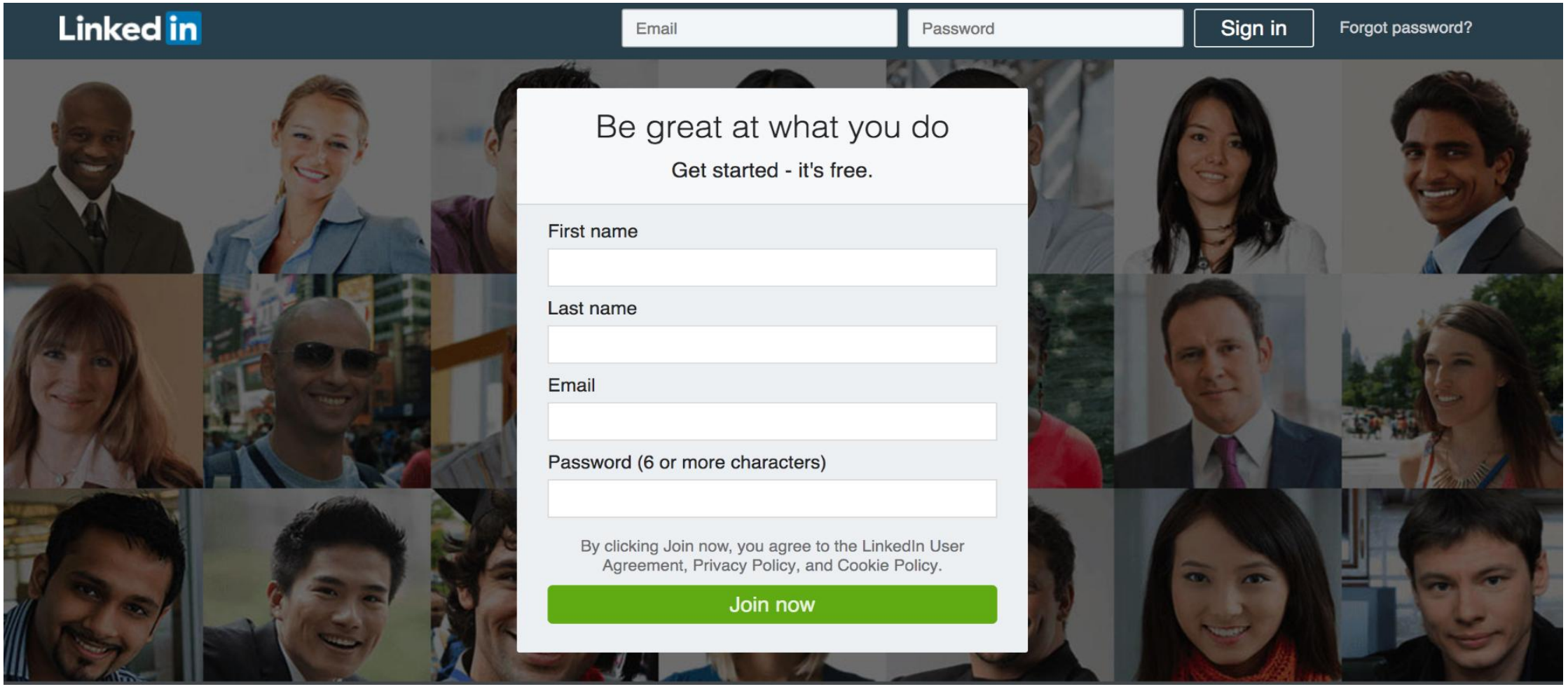
  Then customer should be able to see the shopping cart updated with the product name

  And the product quantity should be 1

- If you see here, there is a keyword And used in the Then example. The keyword And is used when there are more steps for a specific type

Note: Note that every Then statement must be an assertive statement, i.e. must use the keywords such as should, should not, etc.

Example 2 - signing up for a LinkedIn account



Example 2 - Mapping User Stories using BDD

User Story : As a new user (John), I can register a new account on the homepage so that I can access LinkedIn.

GIVEN John is on LinkedIn Registration page

WHEN he enters all the required registration information

AND he hits 'join now'

THEN his LinkedIn account is created

AND he is directed to the profile creation page

AND his confirmation email is sent

Example 3 – Login in to website

```
Feature: Login
  As a new user
  I want to log in to the website
  So that the system can remember my data

  Scenario #1: Successful Log in to the website
    Given A user brings up the login pop-up
    When A user clicks Sign-in
    And A user enters a valid email <email> and password <password>
    And A user clicks Sign-in
    Then A user should be successfully logged into the site

  Scenario #2: Unsuccessful Log in to the website
    Given A user brings up the login pop-up
    When A user enters an invalid email <email> and password <password>
    And A user clicks Sign-in
    Then A user should not be successfully logged into the site
```