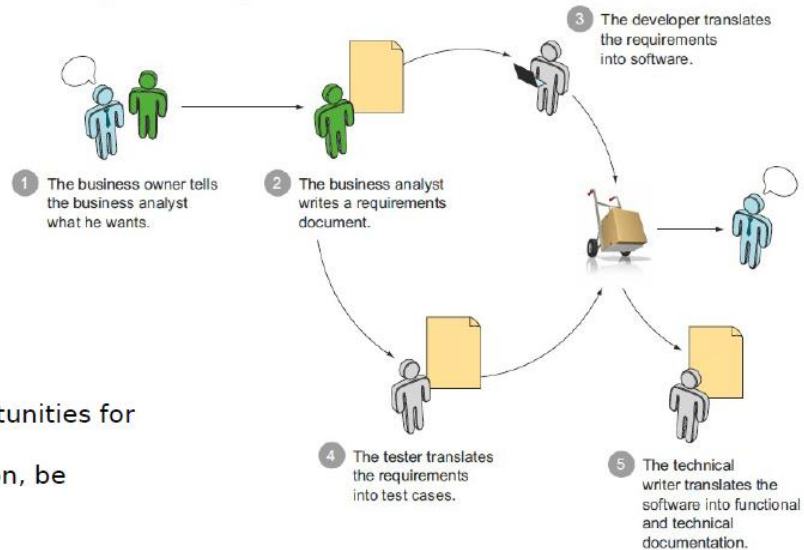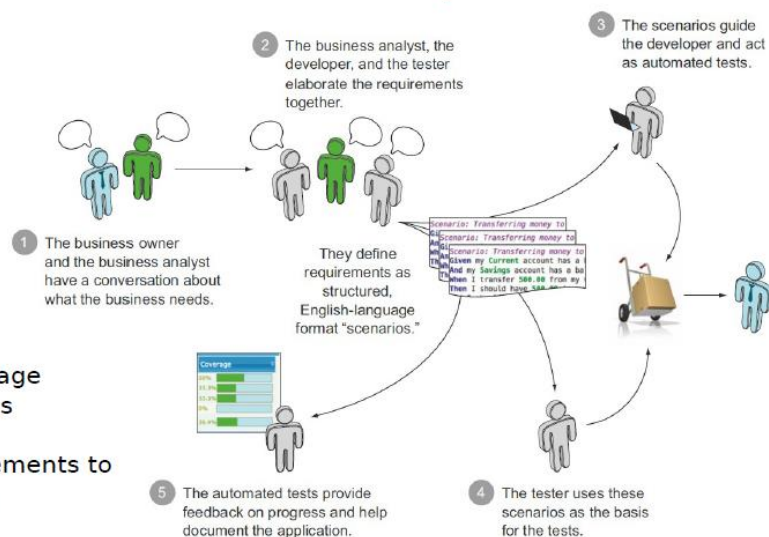# Introductions

Traditional development process

There are many opportunities for information to get lost in translation, be misunderstood, or just be ignored.

1. The business owner tells the business analyst what he wants.
2. The business analyst writes a requirements document.
3. The developer translates the requirements into software.
4. The tester translates the requirements into test cases.
5. The technical writer translates the software into functional and technical documentation.
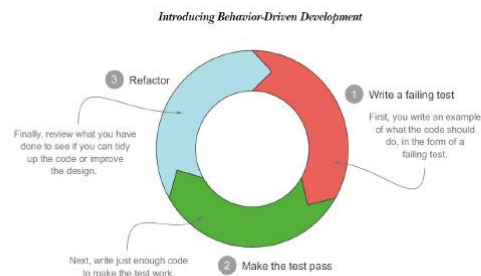

BDD uses conversations around examples

They use a common language that allows for an easy, less ambiguous path from end-user requirements to usable, automatable tests.

1. The business owner and the business analyst have a conversation about what the business needs.
2. The business analyst, the developer, and the tester elaborate the requirements together.

They define requirements as structured, English-language format "scenarios."

3. The scenarios guide the developer and act as automated tests.
4. The tester uses these scenarios as the basis for the tests.
5. The automated tests provide feedback on progress and help document the application.

What is BDD ? What is TDD ?

- A set of software engineering practices designed to help teams build and deliver more valuable, higher quality software faster.

BDD was originally designed as an improved version of TDD

- **Test driven development** (TDD) is a software development approach in which a test is written before writing the code. Once the new code passes the test, it is refactored to an acceptable standard.

Introducing Behavior-Driven Development

3 Refactor

Finally, review what you have done to see if you can tidy up the code or improve the design.

1 Write a failing test

First, you write an example of what the code should do, in the form of a failing test.

Next, write just enough code to make the test work.

2 Make the test pass

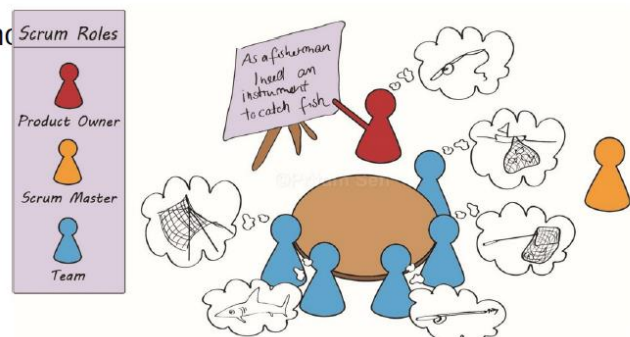## *BDD was originally designed as an improved version of TDD (continue)*

Differentiation between TDD and BDD:

TDD	BDD
focus on the method or function they're testing	focus on what the code should do in business terms.

public class BankAccountTest {
    @Test
    public void testTransfer() {..}
}
	public class BankAccountTest {
	    @Test
	    public void sholud_transfer_funds_to_a_different_bank()
	{..}
	}

WORK TOGETHER TO SPECIFY FEATURES

BDD is a highly collaborative practice:
- Between users and the development team, and within the team itself.
- Business analysts, developers, and testers work together.
- With the end users to define and specify features, and team members draw ideas from

   their individual experience and know-how

What is the benefit/Purpose of this BDD?

- Focus on identifying, understanding, and building valuable features that matter to businesses.
- Deliver real business value
- Viable product
- Teams can deliver more useful features earlier
- Reduced waste
- Reduced costs
- Faster release

What is Cucumber?

- Cucumber is a testing framework which supports BDD.
- It allow us to create **feature file** in plain meaningful English text using a simple grammar defined by a language called **Gherkin**

What is Feature File?

Feature File is an entry point to the *Cucumber* tests. This is a file where you will describe your tests in Descriptive language (Like English).

What is Gherkin ?

- *Gherkin* is a simple, lightweight and structured language which uses regular spoken language to describe requirements and scenarios.

- Is a common language across different domains of project.

Example of Feature file

Feature
- Something that your software does/should do

Feature: <short description>
In order to <business value>
As a <role>
I want <feature>

Feature: GST overview page
*In order to …support GST calculation in the
system*
As a customer
I want to be able to view the GST percentage

Example of Gherkin :

Scenario
- Sequence of steps through the feature
- Independent and isolated. Can run any scenario at any times

Format:
```
Scenario: <concrete example of system behavior>
Given <set up initial state/steps>
When <perform action>
Then <expected outcome/end state>

Scenario: View GST percentage in overview page
Given I login as User
When I go to overview page
Then I should see GST percentage column
```