BDD documentation download

*BDD is a process designed to aid the management and the delivery of software development projects by improving communication between engineers and business professionals. In so doing, BDD ensures all development projects remain focused on delivering what the business actually needs while meeting all requirements of the user.*

TestCases Specifications in Behaviour Driven Development :

Step 1:

Testcases should be defined in a Business level by following a ubiquitous language

Ubiquitous language is a (semi-)formal language that is shared by all members of a software development team — both software developers and non-technical personnel.[15]

Syntax :

In order to(Achieve something)-> Business outcome

As a (Role)

I want to Do this

Example:

In order to Pay Credit Card Payment

As a NetBanking sole owner who have credit access

I want to Navigate to credit card section, Enter amount and process my Payment

Step 2:

Its time to Decide what Test cases are sufficient to certify Business Scenario working

In the software domain, people tend to use diagrams and technical maps. That said, the most common and adopted format for examples is Given-When-Then which is the format you’ll see most used by the BDD community. It follows a very simple structural agreement.

**'Given,** describes the initial context for the example  
**'When'** describes the action that the actor in the system or stakeholder performs  
**'Then'** describes the expected outcome of that action

The primary reason why this format is so popular is because it is the natural way people speak. In many cases, when we ask our clients to provide examples of how they expect a particular capability to be delivered, they naturally describe it in a sentence such as: ‘when I do that, then this should happen’. Here’s an example of an example:

***Scenario: returned items go back to stock***In order to keep track of stockAs a store ownerI want to add items back to stock when they are returned

***Scenario: refunded items should be returned to stock***Given a customer previously bought a black sweater from meAnd I currently have three black sweaters left in stockWhen he returns the sweater for a refundThen I should have four black sweaters in stock

***Scenario: replaced items should be returned to stock***Given that a customer buys a blue garmentAnd I have two blue garments in stockAnd three black garments in stockWhen he returns the garment for a replacement in blackThen I should have three blue garments in stockAnd two black garments in stock

Syntax for Writing Testcases:

Given (what you need to have to perform action), -Prerequisites

When (performs action)- Action

Then (the desired outcome for the user). - Validation

Example:

Given: An account with zero balance

When: I navigate to Credit card Payment Section and click on submit button giving amount

Then: It should throw error message –fund insufficient

Given: An account with sufficient balance who does not have credit card

When: I navigate to Credit card Payment Section

Then: You don't have access error message

Given: An account with sufficient balance who does not have credit card

When: I navigate to Credit card Payment Section and amount

Then: You don't have to access /error message

Advantages:

This can be used as Standard Template where all QA can stick to one common standards of defining Testcases

Each Scenario reflects a Business Value

We can estimate the Test coverage happened for Each Business outcome by going through Test Cases

We can tag these Annotations to Selenium Automation and execute the Business Testcases

Common Standardised Testcase template for both Manual and Automation testing

CUCUMBER INSTALLATION:

* Java
* Environmental var
* Eclipse
* Download Natural plugin in eclipse
* Cucumber project skeleton can be obtained by Maven
* Cucumber supports Quickstart template
* Src/test/java – Cucumber automation test cases
* Pom.xml needs all the jar info
* For cucumber we need, cucumber-java & cucumber-junit.
* What are the main terminologies involved in cucumber automation? -> **Feature***(where u design all ur test cases reg what need to be automated)*, **Step-Definition***(How to automate will be staged here),* **Junit-test-runner***(Connecting these dots)*.
* Always creates packages and file(here is where Given,when format(Gherkin language) is written. Extension for file should be *.feature*) under Src/test/java.
* Install Tidy Gherkin plugin in chrome
* Cucumber options in stepdefinition – features=”path”, glue=”stepDefinition package”