Writing Better BDDs

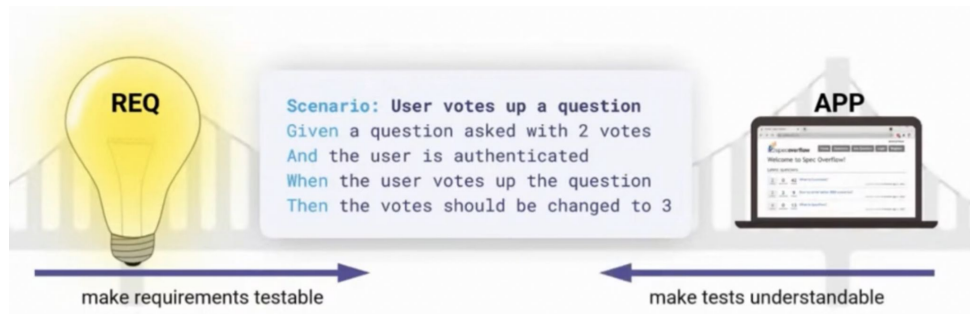## Using B.R.I.E.F Principles

QA Chapter Meetup - Jan 2023
Nihal Alfred

BDD Scenarios

BDD scenarios are a bridge between requirements and the solution.

It makes

- Requirements testable
- Tests understandable

REQ

Scenario: User votes up a question
Given a question asked with 2 votes
And the user is authenticated
When the user votes up the question
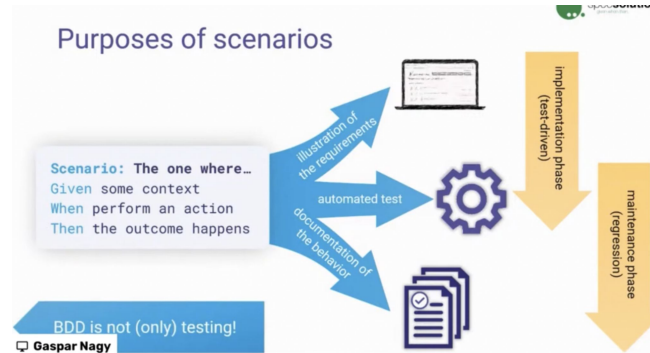Then the votes should be changed to 3

APP

make requirements testable

make tests understandable

Purposes of BDD Scenarios

➔ **Illustration of requirements**

➔ **Automated tests**
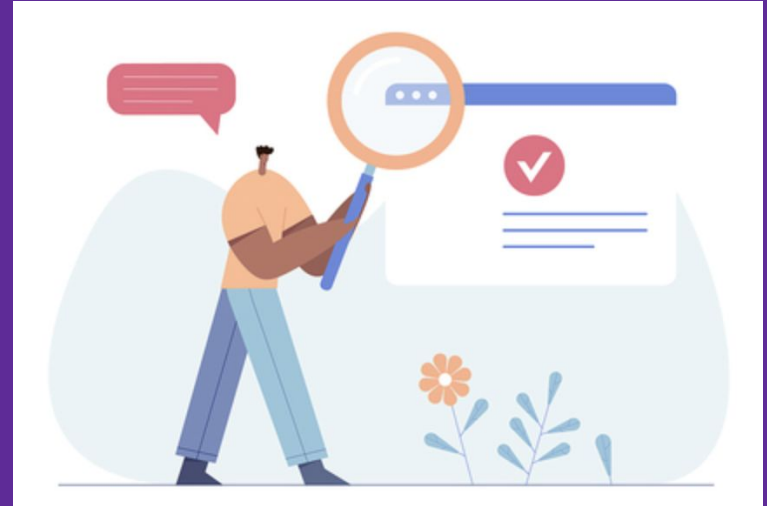
➔ **Documentation of behaviour**

Behaviour Driven Development is about Understanding, Documenting & verifying business requirements through illustrative scenarios.

3 Practices of BDD.

- **Discovery**: Shared understanding is established through collaboration and structured conversations.
- **Formulation**: The examples of system behaviour are documented as scenarios.
- **Automation**: Scenarios are automated to be able to verify the behaviour of the system.

Typical activities done during Formulation:
- **Identify** context, action and outcome (**Given/When/Then**)
- **Review/Fix** business terminology to be used ("**Client**" or "**User**"?)
- Turning scenarios into proper **English sentences**
- **Reuse** existing phrasing styles from other scenarios or even complete steps
- Make sure the steps make a **sequential flow** (so it is easy to automate them)

The Gherkin Syntax

➔ **Business-readable specification language (GWT - introduced by Chris Matt)**

➔ **Developed as part of cucumber but promoted to a tool-agnostic format**

➔ **Designed to be simple, therefore it lacks programing constructs:**

◆ **No conditions**

◆ **No variables**

◆ **No loops**

Anatomy of a scenario - Gherkin

```
Scenario: User votes up a question        ←  Name (title) – identifies the scenario
                                              ("The one where…")
Given a question asked with 2 votes       ←  Given – describes the context
And the user is authenticated             ←  And – continuation of the previous step
When the user votes up the question       ←  When – the action that initiates the behavior
Then the votes should be changed to 3     ←  Then – the expected outcome

     ↑
     The keywords are translated to 70+ languages
```

⚠️ **Note: You should have only one WHEN step**

BDD vs Manual Regression Testing

Similarities:

★ Both target functional quality

★ Both are prescribed ("scripted")

★ Both can be used to catch regression issues

★ Both needs actions to be performed to complete the test

Differences:

★ **Cost factors are different**
  ○ *Manual*: Human execution costs
  ○ *Automated* (BDD): Implementing and maintaining the tests

★ **Verification options are different**
  ○ *Manual*: Human can check many things at once
  ○ *Automated* (BDD): Good in checking only one thing at a time

★ **Testing flows are different**
  ○ *Manual*: Long user journey so that with one execution you can check as many things possible
  ○ *Automated* (BDD): Focus on one business rule and have many smaller tests (scenarios)

QA Chapter Meetup - Jan 2023
Nihal Alfred

```
Scenario: Question Test
Given I navigate to 'http://specoverflow.com/#/'
And wait until link 'Home' becomes visible
Then browser title is 'SpecOverflow - Home'
Given I click on link 'Login'
And wait for 2000 ms
When I click on link 'Ask a question'
And wait for 500 ms
And I clear text box '//input[@type="text"]'
And I type 'Test Question 123' into the text box '//input[@type="text"]'
And I click on element '.ta-text .ta-bind'
And I type 'With some details' into the element '.ta-text .ta-bind'
And I clear text box '(//input[@type="text"])[2]'
And I type 'Gherkin, BDD, dontdothisplease' into the text box '(//input[@type="text"])[2]'
And I click on button '//div[4]/div/div'
And wait for 2000 ms
Then browser title is 'SpecOverflow - Home'
And wait for 1500 ms
And wait until link 'Test Question 123' becomes visible
And the text of element 'a[text()="Test Question 123"]/ancestor::li/div/div[2]/div[2]/div[2]' is
'asked March 22, 2018 by John Doe'
When I scroll to element 'a[text()="Test Question 123"]/ancestor::li'
And I click on link 'Test Question 123'
And wait for 1500 ms
And the text of element 'h1.answer-title.ng-binding' is 'Test Question 123'
And the text of element 'p' is 'With some details'
And I click on link 'Home'
Then browser title is 'SpecOverflow - Home'
```

This scenario is a script and a bad example

Let's make this better

using the B.R.I.E.F principles

B.R.I.E.F

➔ **Business Language**

Business terminology aids cross-discipline collaboration

➔ **Real Data**

Using actual data helps reveal assumptions and edge cases

➔ **Intention Revealing**

Describe the desired outcomes, rather than the mechanism of how they are achieved

➔ **Essential**

Omit any information that doesn't directly illustrate behaviour

➔ **Focussed**

Each scenario should only illustrate a single rule

BRIEF Short scenarios

are easier to **read**, **understand** and **maintain**

QA Chapter Meetup - Jan 2023
Nihal Alfred

Make scenarios focussed by illustrating a single rule

From "How to achieve it" (mechanics) **to "What to expect"** (intention)

Given I have logged in vs adding a tag @authenticated

More details do not necessarily mean more clarity.

Unnecessary, unimportant, irrelevant and incidental details might cause confusion (also bad for maintainability)

Details pushed down to the automation layer

Does Test Question 123 tell a story?

Much Better !!!

Key Facts

Formulation is used to document examples and make them ready for automation

The scenarios should be connected to business rules

Scenarios should follow B.R.I.E.F principles

Further reading

https://git.lystit.com/lyst/app-bdd-scenarios