Specification by Example Explained

- ShriKant Vashishtha
Automation
The first rule of any technology used in a business is that automation applied to an efficient operation will magnify the efficiency. The second is that automation applied to an inefficient operation will magnify the inefficiency.

Bill Gates
Bottleneck, Flow and TOC

the bottleneck or constraint
Testing as Bottleneck
Product Owner as a Bottleneck
What is Scrum?
What is Specification by Example?
Collaborative Way of Getting Requirements and Tests
Best way to Collaborate, around White Board
Acceptance Criteria Example

As a frequent book buyer, I want strong passwords so that my credit card information remains secure

Acceptance Criteria

❖ Test that the system allows passwords with all of the following characteristics:
❖ At least 8 characters
❖ Contains at least one character from each of the following groups:
  ❖ Lower case alphabet
  ❖ Upper case alphabet
  ❖ Numbers
  ❖ Special Characters (!,@,#,$,%,^,&,*)
❖ Test that the system disallows passwords that are lacking any one of the above characteristics.
Any Evolutionary Document Can Become Stale
Welcome to the World of Executable Spec

Scenario 1: Account is in credit
Given the account is in credit
  And the card is valid
  And the dispenser contains cash
When the customer requests €50
Then
  ensure the account is debited €50
  And ensure €40 is dispensed
  And ensure the card is returned
scenario "a valid person has been entered", {
    when "filling out the person form with a first and last name", {
        selenium.open("http://acme.racing.net/greport/personracereport.html")
        selenium.type("fname", "Britney")
        selenium.type("lname", "Smith")
    }

    and "the submit link has been clicked", {
        selenium.click("submit")
    }

    then "the report should have a list of races for that person", {
        selenium.waitForPageToLoad("5000")
        values = ["Mclean 1/2 Marathon", "Reston 5K", "Herndon 10K", "Leesburg 10K"]
        for(i in 0..<values.size()){
            selenium.getText("//table//tr[${(i+3)}]/td").shouldBeEqualTo values[i]
        }
    }
}
Let’s Dig Deeper
Specifications with examples

Customer who buy three books gets free shipping

<table>
<thead>
<tr>
<th>Number of books</th>
<th>free shipping</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>Yes</td>
</tr>
</tbody>
</table>
You are aiming for key examples enough to understand the function.

Not necessary to have every single example.
Another Scenario and Example

Scenario: To get a loan you need to be older than 18

<table>
<thead>
<tr>
<th>Age of applicant</th>
<th>Can take loan?</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>-</td>
</tr>
<tr>
<td>18</td>
<td>X</td>
</tr>
<tr>
<td>19</td>
<td>X</td>
</tr>
<tr>
<td>null?</td>
<td>eeeeh?</td>
</tr>
</tbody>
</table>
Good test may not be necessarily right candidate to understand the feature.
Specifications vs Automated Tests

Automated Tests

Specifications
Aim of Specifications

- Shared Understanding
- Testing
- Documentation
Characteristics of Specifications

• Examples should be precise and testable
• Scripts are not specification (not about how the system is built. click on this button, move to next page. It’s not about UI)
• Specification should be about business functionality, not software design
• Specifications should be self-explanatory → shouldn’t need to cross-reference a lot of sources. Only specification document should be sufficient to know about the system
• Specifications should be in domain language
Specification Styles

1 **Feature**: Google Search
2 As an Internet user I want to use Google for searches
3
4 **Scenario**: Search for ATDD
5 **Given** the default Google page
6 **When** I search for 'ATDD'
7 **Then** I will find lots of results

<table>
<thead>
<tr>
<th>Parking costs for</th>
<th>Valet Parking</th>
<th>parking duration</th>
<th>parking costs?</th>
</tr>
</thead>
<tbody>
<tr>
<td>30 minutes</td>
<td></td>
<td>$ 12.00</td>
<td></td>
</tr>
<tr>
<td>3 hours</td>
<td>$ 12.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 hours</td>
<td>$ 12.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 hours 1 minute</td>
<td>$ 18.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12 hours</td>
<td>$ 18.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24 hours</td>
<td>$ 18.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 day 1 minute</td>
<td>$ 36.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 days</td>
<td>$ 54.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 week</td>
<td>$ 126.00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Specification Styles

Keyword Driven Style

1 ...
2 2 Invalid combinations
3
4 |scenario |invalid combination|firstLight||secondLight|
5 |set first light |@firstLight
6 |set second light|@secondLight
7 |execute
Feature: Transferring money between accounts
   In order to manage my money more efficiently
   As a bank client
   I want to transfer funds between my accounts whenever I need to

Scenario: Transferring money to a savings account
   Given my Current account has a balance of 1000.00
   And my Savings account has a balance of 2000.00
   When I transfer 500.00 from my Current account to my Savings account
   Then I should have 500.00 in my Current account
   And I should have 2500.00 in my Savings account

Scenario: Transferring with insufficient funds
   Given my Current account has a balance of 1000.00
   And my Savings account has a balance of 2000.00
   When I transfer 1500.00 from my Current account to my Savings account
   Then I should receive an 'insufficient funds' error
   Then I should have 1000.00 in my Current account
   And I should have 2000.00 in my Savings account
Gherkin Primer

The natural order of a scenario is *Given ... When ... Then:*
- *Given* describes the preconditions for the scenario and prepares the test environment.
- *When* describes the action under test.
- *Then* describes the expected outcomes.

The *And* and *But* keywords can be used to join several *Given, When, or Then* steps together in a more readable way:

Given I have a current account with $1000
And I have a savings account with $2000

Several related scenarios can often be grouped into a single scenario using a table of examples. For example, the following scenario illustrates how interest is calculated on different types of accounts:

**Scenario Outline: Earning interest**
- Given I have an account of type `<account-type>` with a balance of `<initial.balance>`
- When the monthly interest is calculated
- Then I should have earned at an annual interest rate of `<interest-rate>`
- And I should have a new balance of `<new-balance>`

**Examples:**

<table>
<thead>
<tr>
<th>initial-balance</th>
<th>account-type</th>
<th>interest-rate</th>
<th>new-balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000</td>
<td>current</td>
<td>1</td>
<td>10008.33</td>
</tr>
<tr>
<td>10000</td>
<td>savings</td>
<td>3</td>
<td>10025</td>
</tr>
<tr>
<td>10000</td>
<td>supersaver</td>
<td>5</td>
<td>10041.67</td>
</tr>
</tbody>
</table>
Gherkin Primer

You can also use the following tabular notation within the steps themselves in order to display test data more concisely. For example, the previous money-transfer scenario could have been written like this:

Scenario: Transferring money between accounts within the bank

- Given I have the following accounts:
  - account | balance |
  - current  | 1000     |
  - savings  | 2000     |

- When I transfer 500.00 from current to savings

- Then my accounts should look like this:
  - account | balance |
  - current | 500      |
  - savings | 2500     |
Efficacy of Test Automation
Test Pyramid

UI (10%)

Service (20%)

Unit (70%)
How much web testing do you really need?

• Automated acceptance criteria, related to business rules or calculations, are more effectively done directly using the application code rather than via the user interface.

• You only need a web test for two things:
  • Illustrating the user’s journey through the system
  • Illustrating how a business rule is represented in the user interface

• A good rule of thumb is to ask yourself whether you’re illustrating how the user interacts with the application or underlying business logic that’s independent of the user interface.

• For example, suppose you were testing a user authentication feature. The acceptance criteria might include the following:
  • The user should receive feedback indicating the strength of the password entered.
  • Only strong passwords should be accepted
By the way who owns the tests?
Collective Test Ownership
Thank You

www.agilebuddha.com, www.malonus.in
@LinkedIn - https://www.linkedin.com/in/shrikantv/
@vashishthask