

Specification by Example: a Love Story

Author:	Alister Scott
Blog	alisterbscott.com
Date	18 May 2011

Background Information

Janet and Dave own 'Beautiful Tea': a boutique tea estate in the Byron Bay hinterland in Australia. They grow and sell organic loose leaf tea direct to tea connoisseurs using an online ordering application developed a number of years ago. The online ordering application was originally developed and supported by an external company, but was transitioned in house a few years ago due to poor service and prohibitive cost. In transitioning it in house, Dave and Janet set up a small team to develop and support the online ordering application:



Beautiful Tea Staff

Janet and Dave: Owners & Subject Matter Experts

Henry: Customer Support

Mark: Business Analyst

Madison: Tester

Monique: Programmer

Quality & Velocity Issues

Ever since it was created, the Beautiful Tea online ordering application has been plagued with quality and velocity issues. These problems have meant not only losing money directly (for example miscalculating shipping and GST) but have also hindered business expansion plans. For example, Janet and Dave have had a lot of demand from small tea specialist retailers to resell their product, but the online ordering application is so brittle and difficult to change, they have not been able to make the necessary changes to facilitate this.

The website system documentation provided by the external company was not only sparse, but confusing to read and lacking in detail. The only truth about the system *is* the system, so this means complex system archeology is needed to solve a simple problem, which takes a large amount of Monique's time, and means less changes can be added in short periods of time.

Phase One: Automated Testing 101

Due to ongoing quality issues with introducing any change, Madison, the tester for Beautiful Tea, embarked on a mission to create some automated web tests using Selenium.

Example Automated Test Script

```
open "http://beautifultea.com"
assertTitle "Beautiful Tea"
pause "2000"
clickAndWait "link=Teas"
pause "3000"
assertTitle "Range of Teas"
clickAndWait "link=Byron Breakfast"
assertTitle "Byron Breakfast"
click "//input[@name='buy' and @value='1']"
type "quantity", "6"
clickAndWait "//input[@value='Add to cart']"
assertTitle "Beautiful Tea Cart"
```

Whilst Madison got the Selenium scripts running fairly quickly, mostly recording each script using Selenium IDE, problems began to surface quickly:

- Due to the large number of test scripts generated, a small change in the application meant lots of tests no longer worked, and these scripts took hours to run;
- Test data was hard coded in the test scripts, therefore Madison found it hard to run the tests in different environments;
- Each script was written in code, and therefore owners Janet and Dave had difficulty understanding the intention of each script, especially when the script included XPath and CSS selectors; and
- Madison was the only person in the team that fully understood the testing suite.

Phase Two: Cucumber 101

Madison began noticing the problems associated with the scripted approach to automated testing, and started to read a lot about Cucumber, a BDD style framework that was becoming increasingly popular in the agile testing world. With nothing to lose, Madison began converting her existing Selenium scripts to Cucumber scenarios and step definitions.

Example Cucumber Feature

Feature: Beautiful Tea Shipping

Scenario: Free shipping in Australia

Given I am on the Beautiful Tea home page
When I search for 'Byron Breakfast' tea
Then I see the page for 'Byron Breakfast' tea
When I add 'Byron Breakfast' tea to my cart
And I select 10 as the quantity
Then I see 10 x 'Byron Breakfast' tea in my cart
When I select 'Check Out'
And I enter my country as 'Australia'
Then I see the total including GST
And I see that I am eligible for free shipping

Scenario: No free shipping outside Australia

Given I am on the Beautiful Tea home page
When I search for 'Byron Breakfast' tea
Then I see the page for 'Byron Breakfast' tea
When I add 'Byron Breakfast' tea to my cart
And I select 10 as the quantity
Then I see 10 x 'Byron Breakfast' tea in my cart
When I select 'Check Out'
And I enter my country as 'New Zealand'
Then I see the total without GST
And I see that I am not eligible for free shipping

Scenario: No free shipping in Australia

Given I am on the Beautiful Tea home page
When I search for 'Byron Breakfast' tea
Then I see the page for 'Byron Breakfast' tea
When I add 'Byron Breakfast' tea to my cart
And I select 1 as the quantity
Then I see 1 x 'Byron Breakfast' tea in my cart
When I select 'Check Out'
And I enter my country as 'Australia'
Then I see the total including GST
And I see that I am not eligible for free shipping

It took some time to convert these scripts across Cucumber scenarios, but after a few iterations Madison felt these were working well.

It was only when a reasonably sized change was planned to the entire purchase process that Madison realized the scope of change needed to her Cucumber scenarios. Madison organized a whole team workshop a few days before a new iteration to discuss the changes needed to the Cucumber features. Mark, the business analyst, as well as Janet and Dave as the subject matter experts quickly pointed out how hard it was to understand the differences between the various scenarios. They also found some of the terminology used didn't match what was used around their tea estate. Madison also realized she essentially she needed to package herself with the Cucumber scenarios to often explain what was happening, which isn't what she wanted, so these scenarios needed some work.

Phase Three: Shared Understanding: moving towards Executable

Specifications

Madison did some research and found out some principles that could help her improve the cucumber scenarios she had written:

- **Specifications, not scripts:** she should move to less workflow based scenarios but more specifications about what is needed, as these are easier to understand, more precise and testable;
- **Abstract:** the specification should be abstract enough to highlight the detail, remove the noise, and not being tied to the implementation of the user interface;
- **Ubiquitous language:** the language used by the team and specifications should be consistent throughout the development process to ensure a shared understanding;
- **Edge cases:** unusual variances should be specified to ensure clarity of expectations: “things that seem obvious kill us, if something sounds obvious, that’s where the danger is”;
- **Key examples:** each decision point should have 5 - 6 key examples, and not more, so it is clear what is expected. These can be created by focussing on the differences between existing scenarios;
- **End-to-End flows:** only a few (1-3) end-to-end flows, not a combination of every decision point combination
- **Accessible:** publishing the specifications so Janet, Dave and others can easily access the latest versions.

With this arsenal of tricks, Madison organized a workshop to rewrite the cucumber *tests* as *specifications*.

Example Specification

Feature: Beautiful Tea Shipping Costs

- Australian customers pay GST
- Overseas customers don't pay GST
- Australian customers get free shipping for orders \$100 and above
- Overseas customers all pay the same shipping rate regardless of order size

Scenario: Calculate GST status and shipping rate

Given the customer is from <customer's country>

When the customer's order totals <order total>

Then the customer <pays GST>

And they are charged <shipping rate>

Examples:

customer's country	pays GST	order total	shipping rate
Australia	Must	\$99.99	Standard Domestic
Australia	Must	\$100.00	Free
New Zealand	Must Not	\$99.99	Standard International
New Zealand	Must Not	\$100.00	Standard International
Zimbabwe	Must Not	\$100.00	Standard International

Straight away the team could look at the specification and see it was clear about shipping costs and GST status, something that had been confusing before and had cost Beautiful Tea money in production issues.

Phase Four: Living Documentation

Now that collaboration had begun on creating specifications, these became more and more of the focus point for any change. Often Madison would use ad-hoc conversations to collaborate and evolve these specifications, as well as 'three amigos' sessions, consisting of a BA/SME, a tester and a programmer. Everyone began feeling responsible for quality.

Madison found the specifications and associated acceptance tests much easier to maintain as everyone understood them and they were less connected to the actual online ordering implementation.

Henry found the specifications vital in his role as customer support, as he had one source of truth, and Monique the programmer found herself performing less and less system archeology to answer a simple question about what the online ordering system actually did.

Most importantly, as there were always up to date executable specifications, this meant it was much easier to update the system to support the business without the risk of introducing unintended issues. Owners Janet and Dave were able to specify the new online ordering functionality for resellers and this was easily incorporated into the application by the team. Delivering the new functionality quickly and without issue meant the business was able to grow.

The team had fallen in love in Specification by Example.