

Specification by Example



BEHAVIOR DRIVEN TEST DEVELOPMENT

How Did We Get Here?



- Test Driven Development focused developers on writing test code before writing code.
- *Problem – Well tested code was not meeting the customers expectations.*
- Behavior Driven Development is a process created by Dan North in 2003 as an extension of Test Driven Development in order to provide a more accurate way of taking user story features and translating them into requirements.
- *Solution – Provide a means of describing the behavior of the feature in non-technical terms.*

Behavior Driven Development

What are the goals of BDD?

- Drive application development based upon Business Value.
- Write tests before you write code.
- Work in small increments of work.
- Provide the development team an ability to refactor code quickly.
- Designed to illustrate the behavior of a feature.

What are the Benefits?

- Reduce the time for code implementation.
- Specifying via example will lead to more modularized, flexible, and extensible code.
- Encourages close teamwork between the entire Project Team.

4

Specification by Example

• What's the Process?

- Identify Acceptance Criteria (Examples) that supports a User Story/Feature before any code is written.
- Have a complete review of the Examples and obtain sign off from the team before starting coding.
- Coding is complete, and a feature can be implemented, when all the Acceptance Criteria (aka Examples) pass.

• What are the Benefits of Examples?

- Creates business value, because only what is needed to support the feature is coded, nothing more, nothing less.
- Quality increases when developers understand how the feature will be used.
- Sprint quality improves with Agile/BDD zero defect policy.

5

Specification by Example

- Examples = Acceptance Criteria
- Building Examples
 - Behavior Driven Development provides the following process for defining a testable feature:
 - Given** <some precondition>
 - And** <additional preconditions
 - When** <an action/trigger occurs>
 - Then** <some post condition>
 - And** <additional post conditions>
- Use **And** to provide further context for the feature.
- Similar to Use Case development

6

Specification by Example

- **What's the Story?**
- **Review User Stories in order to create a Test Scenario Outline**
 - Why? To confirm that you understand stories sufficiently in order to identify tests that will support them.
- **Example-** *As a customer, I want to withdraw cash from an ATM, so that I don't have to wait in line at the bank.*
- **I. Use short descriptive names for your Test Scenarios:**
 - **Example: Scenario 1** - Account is in Credit
- **II. Each Scenario should represent a Different Facet of the User Story**
 - **Example: Scenario 2** – Account is Overdrawn
- **III. Identify All the Variables that need to be supported in the User Story**
 - What are considered customers (personal, business, corporate)?
 - What are considered accounts (savings, checking, credit)?

7

Specification by Example

What's the Outline do for me?

- Describes the Scenario in plain language - No one remembers what Scenario 1.6.1 is. Speak in the language of the business domain.
- Encourages conversations
- Assists with Leveling the Story/Feature (Is it too big, too small, inclusive)
- Identifying potential New Stories/Features, because of project deliverable timelines.
 - ✦ Not supporting Business Accounts in the 1st Release.
 - ✦ Not supporting Savings Accounts in the 1st Release.

8

Specification by Example

Given/When/Then – Building The Behavior of Your Test First

- In order to develop good Examples for your tests, you need to first build out your test using the Behavior Driven Development process:
 - Given** - some initial context
 - Example: User Type
 - Set Up any Preconditions (use And)
 - When** – An Event Occurs
 - Submit a page, click a link, select an option
 - Then** – The Expected Response
 - And** – Use with Given, When or Then when multiple steps need to be executed.
- This process allows the development of rich contextual tests relatively quickly.

9

Specification by Example

- **Building the Test – Given and Preconditions**

Feature/Story:	<i>As a customer, I want to withdraw cash from an ATM, so that I don't have to wait in line at the bank.</i>
Scenario 1	Account is in Credit
Given Statement	Given I'm a customer with a valid account And My Account is in Credit
Scenario 2	Account is Overdrawn
Given Statement	Given I'm a customer with a valid account And My Account is Overdrawn

10

Specification by Example

- **What to be looking for:**

- Identify steps that can be used in other scenarios and/or User Stories/Features where possible.
 - × Why? – Building a set of repeatable steps with your code reduces the amount code the team has to maintain.
- Add preconditions to the steps to further define the steps to the action, don't try to include everything in a single Given statement.
 - × Use parameters <> to identify different paths the test can take:
 - Given I am a <User> (could be Consumer, Business, etc..)
 - Note - Anything you put in a Parameter becomes a separate test in your example table.

11

Specification by Example

- **Building the Test – The Action - When**

User Story:	<i>As a customer, I want to withdraw cash from an ATM, so that I don't have to wait in line at the bank.</i>
Scenario 1	Account is in Credit
Given Statement	Given I'm a customer with a valid account And My Account is in Credit
When Statement	When I withdraw money from my account
Scenario 2	Account is Overdrawn
Given Statement	Given I'm a customer with a valid account And My Account is Overdrawn
When Statement	When I withdraw money from my account

12

Specification by Example

- Using the **When** Statement Appropriately:
 - Keep the **When** statement to one specific action (step) unless the process clearly has more than one step before completion. Why?
 - If the When statement has many steps then the test potentially has complexity that isn't clearly understood.
 - Makes it more difficult to build your Example table with multiple actions, tests are cleaner when they can be broken down to a single Action.
 - Ensure that the When statement is not a Precondition:
 - Example: When there is cash in the ATM dispenser AND I withdraw Cash from My Account.
 - "cash in the dispenser" is not an action and can be removed as a pre-condition in the Given statement

13

Specification by Example

- **Building the Test – What's the Outcome?**

User Story:	<i>As a customer, I want to withdraw cash from an ATM, so that I don't have to wait in line at the bank.</i>
Scenario 1	Account is in Credit
Given Statement	Given I'm a customer with a valid account And My Account is in Credit
When Statement	When I withdraw money from my account
Then Statement	Then my account should be debited And requested amount should be dispensed
Scenario 2	Account is Overdrawn
Given Statement	Given I'm a customer with a valid account And My Account is Overdrawn
When Statement	When I withdraw money from my account
Then Statement	Then I should receive an error message And requested amount should not be dispensed

14

Specification by Example

- **The Then Step**
- Keep the **Then** statements to the Expected Result/Outcome of the Action Identified in the **When** statement.
- Use the **And** statement to identify more than one outcome from the **When** Action
- Keep Assertions for Different Elements on Separate Lines
- **Why?**
 - Provides Clarity regarding the expected behavior of the outcome.
 - Leads to clear definition for your automation scripts.

15

Specification by Example

- Using Parameters and Example Patterns
 - I have multiple tests that I can identify for my scenarios, do I have to write out a scenario for each one?
 - ✖ No - You can use parameters and example tables to test all the permutations of a scenario.
 - Where Can I use Parameters?
 - ✖ Parameters can be located in any of the Given, or Then Steps
 - ✖ Parameters can be used in the When statements provided it isn't used to alter the actual action. (Example: withdraw money to deposit money)
 - How are Parameters Identified in the Steps?
 - ✖ Parameters are identified by enclosing a parameter name within carets
 - ✖ Example: Given I am a <user> - In your business requirements the types of users should be identified if there is more than one. Each user can be setup in the Example table.

16

Specification by Example

- Using Parameters and Example Patterns for Permutations
- What are the guidelines for parameter naming conventions?
 - Parameter names should be meaningful
 - Parameter names should be all lower case
 - Use “_” to concatenate parameters with more than 1 descriptive word
- Can I use trigger words to differentiate an expected result?
 - Yes, trigger words like (should, should not), (will, will not), can be used as a parameter with scenarios that have multiple permutations.
 - Use “should?” (question mark at the end of the trigger word) to identify it as a trigger word.
- Let's rewrite our Scenario using Parameters
 - Customer types include (Customer, Small Business and Corporate)
 - Account types include (Checking and Savings)

17

Specification by Example

- **Using Parameters and Example Patterns for Permutations**

User Story:	<i>As a customer, I want to withdraw cash from an ATM, so that I don't have to wait in line at the bank.</i>
Before	After
Given I'm a customer with a valid checking account And My checking account is in Credit When I withdraw money from my checking account Then my checking account should be debited And the requested amount should be dispensed	Given I'm a <user_type> And I have a valid <account_type> And my <account_type> is in credit When I withdraw money from my <account_type> Then my <account_type> account should be debited And the requested amount should be dispensed

18

Specification by Example

- **Using Parameters and Example Patterns for Permutations**

- Create an Example table with all the Data Permutations
- Example tables should be below the Scenario with "Example:" as the header
- Use the Pipe character as delimiters for the columns
- Parameter names are the Column Headers

Given I'm a <user_type> And I have a valid account type <yes/no> And my <account_type> account is in Credit When I withdraw money from my account Then my should be debited And the requested amount should be dispensed			
Example:			
user_type	yes/no	account_type	
Consumer	yes	Checking	
Consumer	yes	Savings	
Small Business	yes	Checking	
Small Business	yes	Savings	
Consumer	no	new account page	

19

Specification by Example

- I'm Done, Now What Do I Do?
- **Where Should I Store Acceptance Criteria?**
 - There are many ways that Acceptance Criteria can be managed, things to think about:
 - ✦ Documentation should be lightweight
 - ✦ Easily accessed by individuals outside of the team
 - ✦ Tools such as Fitnesse, Rubymine and Cucumber provide a great place to store and maintain the tests via version control systems.
- **Who Needs to Review the Scenarios?**
 - Business Analysts, Product Manager and Developers are the key roles that should review and approve Acceptance Criteria.

20

Specification by Example

- **What is the Purpose of the Review?**
 - Enables additional conversations between BA, QA and Development
 - Make final modifications to the Acceptance Criteria
- **What Happens When the Acceptance Criteria is Approved?**
 - Acceptance Criteria is considered in a final state and ready for development to start.

21