ADOPT BDD AND DELIVER WHAT MATTERS!!

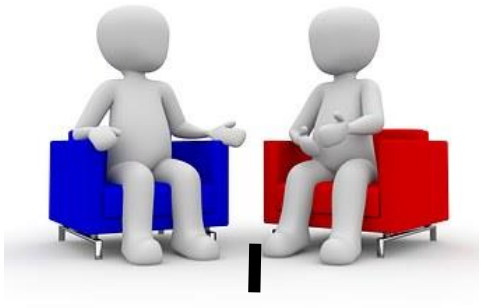
By:
Ruchi Chopra

Contents

- Background on why we need BDD
- What is BDD and its benefits
- BDD and Requirements
- Automate Acceptance Tests
- Testers Role in BDD

Basic Requirements Flow

Business Talks to Customers and collect Requirements



Is it possible for the customer to explain all the needs upfront????

Business thinks of the required solution and communicates it to technical team who implement it.

Can all the business outcomes and technical complexities be understood upfront????

Technical Team implements the solution based on what business suggests.

Does the technical team understood well what business expects????

Lets Take an Example ..

Customer requirement

1. Have one trunk
2. Have four legs
3. Should carry load both passenger & cargo
4. Black in color
5. Should be herbivorous

Is the Solution really delivering what the customer wanted????

Basic Requirements Flow When Practising BDD

Continuous feedback as milestones are achieved

Business and Technical Team in Continuous discussion around the expected solution

What is BDD??

BDD which stands for ***Behaviour Driven Development*** is a set of software engineering practices which help to deliver faster while meeting customer expectations.

Benefits of BDD:

- ✓Reduces waste
- ✓Reduce costs
- ✓Faster releases
- ✓Easier and safer changes

Key Characteristics of a Team Practising BDD

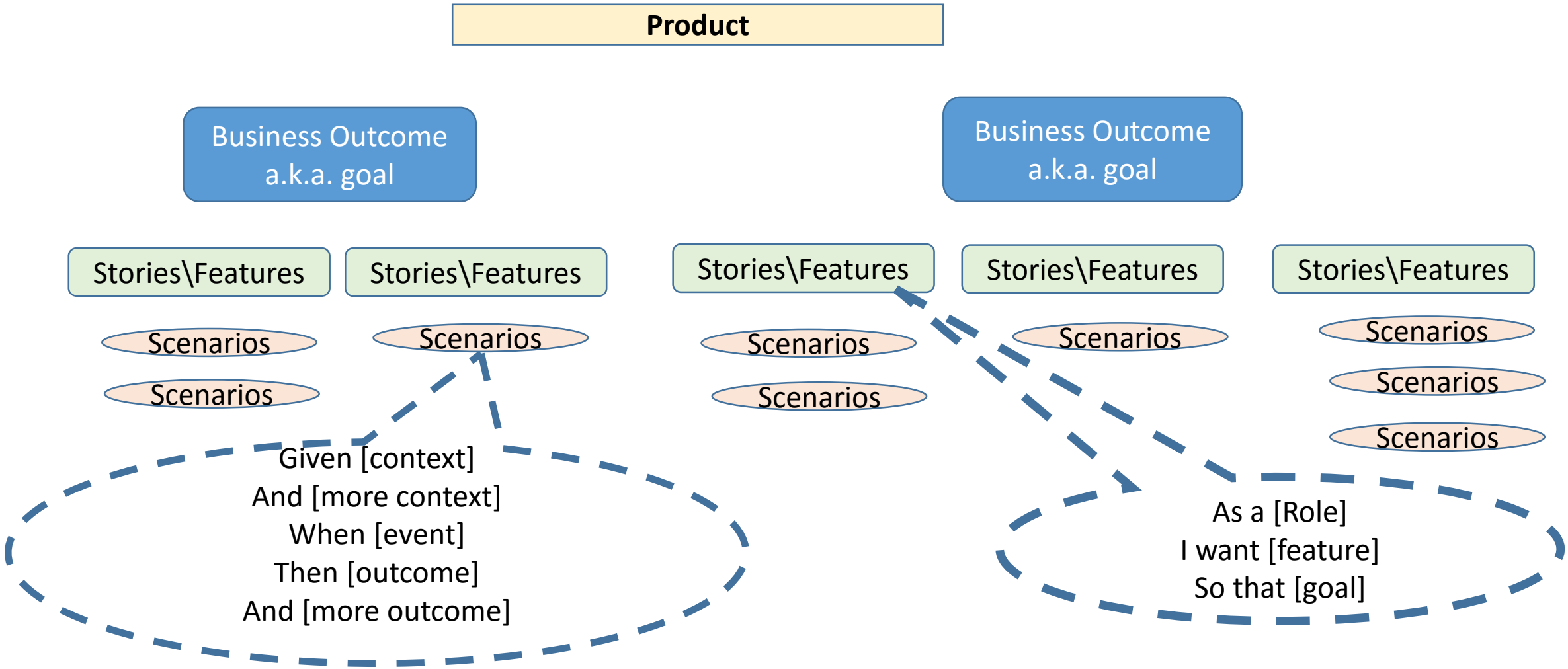
- ✓Requirements are described as user scenarios.
- ✓Automated acceptance tests are validating the user scenarios.
- ✓Automated acceptance tests are acting as live specification and demonstrate the covered milestones.
- ✓Great collaboration between business, developers, testers and a blur line in the role of business analysts, developers and testers.
- ✓Gets continuous feedback from customers.

BDD is seen as more effective in teams practising Agile.

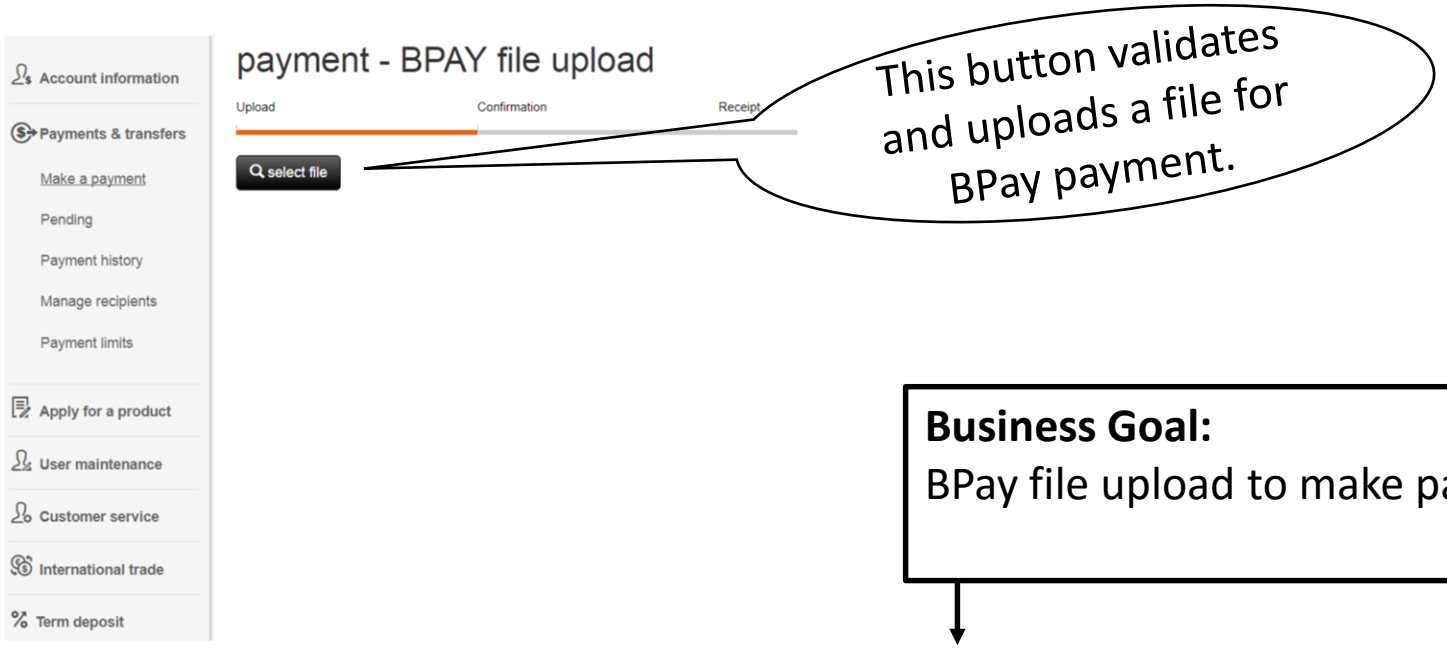
BDD and Requirements

Behaviour-driven development is an **“outside-in”** methodology. It starts at the outside by identifying **business outcomes**, and then drills down into the **feature set** that will achieve those outcomes. Each feature is captured as a “**story**”, which defines the scope of the feature along with its **acceptance Criteria's**.

Requirements Structure Following BDD



Lets Take a Real Example....



Business Goal:
BPay file upload to make payments.

Story 1:
As a User
I should be able to validate my BPay File
So that I can fix the errors before payment

Scenario 1:
Given User is on payment- BPay File Page
When he uploads a file having BPay code error
Then error with error message "you have error in BPay Code" is displayed

Scenario 2:
Given User is on payment- BPay File Page
When he uploads a file having CRN code error
Then error with error message "you have error in CRN Code" is displayed

Story 2:
As a User
I should be able to Upload a BPay File
So that I can make payments

Scenario 1:
Given User is on payment- BPay File Page
When he clicks select file button to upload a csv file
Then the file is uploaded without errors.

Automated Acceptance Tests

Automated Acceptance Tests are written to verify scenarios (Acceptance Criteria's)

Key characteristics of Automated Acceptance Tests:

- ✓ Automated Acceptance Tests are written for all the requirements whether functional or non functional.
- ✓ The tests follow the Given-When-Then format.
- ✓ They test the end to end Business Logic as it will be used in production.
- ✓ They act as Regression Tests as well as the live Specification.
- ✓ They prove that software is meeting customer requirements.
- ✓ Shared test reports give confidence to the stakeholders for project deliveries and achieved milestones.

An Example..

```
_include('../../../../setup-test.js');
_include('../page-functions/payment-bpay-file-upload-page.js');
_include('../page-functions/make-a-payment-page.js');
_include('./BPayBatchCommonFunctions/BpayBatchLogInAndSelectAFile.js');

var $BPBPans = JSON.parse(_readFile("./TestData/BPayBatch_Upload_Pans.json"));

new Scenario("User uploads a file with BPay Code Error");

new Given("User is on payment- BPay File Page");

$bpayFileUploadPage = new BpayFileUploadPage();

new When("he uploads a file having BPay code error");

$bpayFileUploadPage.uploadFile("\\TestData\\500trans_1ToSignAccount_valid.csv");

new Then("error with error message “you have error in BPay Code” is displayed");

$bpayFileUploadPage.validateBPayCodeErrorExists();

logout();
```

Test implementation for Acceptance Criteria

An Example..

```
using PrimeOneLib;
using TestStack.BDDfy;
using NUnit.Framework;

namespace IBSPayments.TestSuite
{

    public class IBSPaymentsPost : BaseTest
    {
        private readonly string _request = "<XmlRequest><Header><RequestName>IBSPaymentPost</RequestName><UserID>CBSTEST</Us
        private string _response;

        [Test()]
        public void IBSPaymentPost_Send10Transactions()
        {
            this.Given(_ => GivenUserisOnBPayFileUploadPage(), "User is on payment- BPay File Upload Page")
              .When(_ => WhenFileHavingBpayCodeErrorisUploaded(), "he uploads a file having BPay code error")
              .Then(_ => ThenBpayErrorisDisplayed(), "error with error message “you have error in BPay Code” is displayed")
              .BDDfy("User uploads a file with BPay Code Error");

        }

        #region BDDfy methods
        private void ThenBpayErrorisDisplayed()
        {
           //Assert.IsTrue(_response.Contains("Processed"));
```

Test implementation for Acceptance Criteria

Automated Test Reporting Acting as Live Specification

Summary:

Stories	0	
Scenarios	1	
Passed	1	☑
Inconclusive	0	☑
Not Implemented	0	☑
Failed	0	☑

Details:

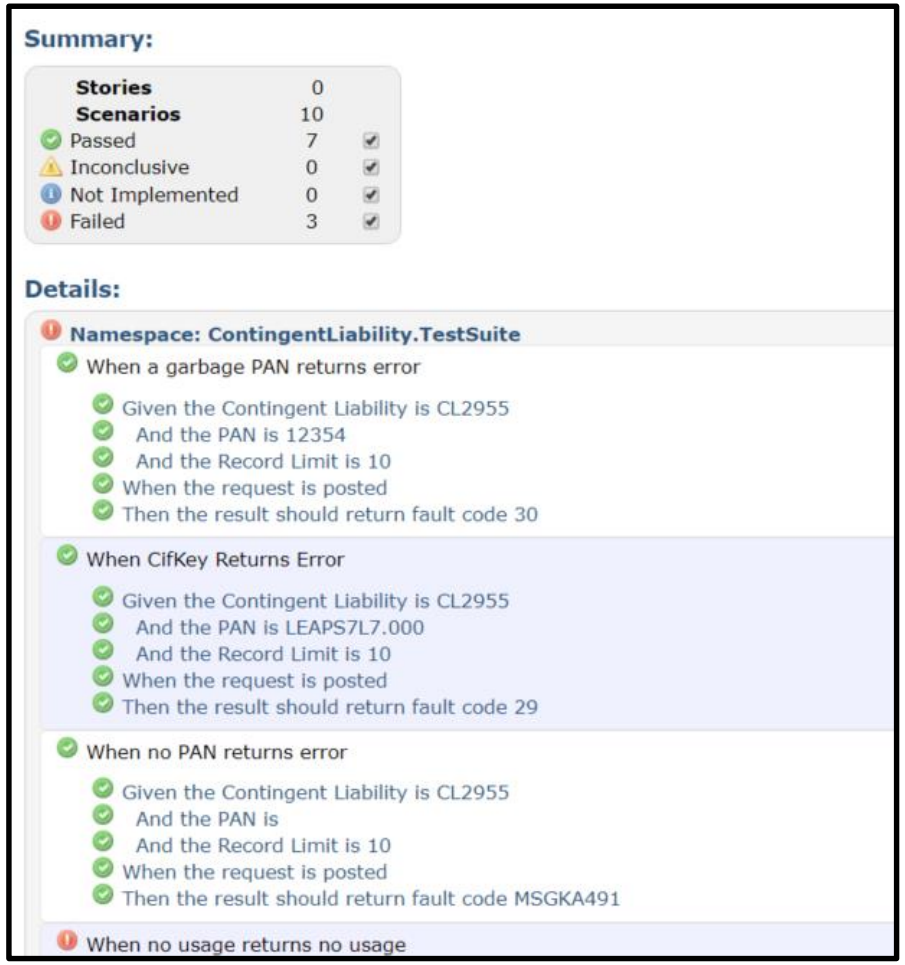
Namespace: IBSPayments.TestSuite

User uploads a file with BPay Code Error

- User is on payment- BPay File Upload Page
- he uploads a file having BPay code error
- error with error message "you have error in BPay Code" is displayed

Tested at: 7/11/2017 11:22:36 PM

Automated Test Reporting Acting as Live Specification



Summary:

Stories	0	
Scenarios	10	
Passed	7	☑
Inconclusive	0	☑
Not Implemented	0	☑
Failed	3	☑

Details:

Namespace: ContingentLiability.TestSuite

When a garbage PAN returns error

- Given the Contingent Liability is CL2955
- And the PAN is 12354
- And the Record Limit is 10
- When the request is posted
- Then the result should return fault code 30

When CifKey Returns Error

- Given the Contingent Liability is CL2955
- And the PAN is LEAPS7L7.000
- And the Record Limit is 10
- When the request is posted
- Then the result should return fault code 29

When no PAN returns error

- Given the Contingent Liability is CL2955
- And the PAN is
- And the Record Limit is 10
- When the request is posted
- Then the result should return fault code MSGKA491

When no usage returns no usage

Testers Role in a Team Practising BDD

- To ensure Requirements are written in User's Perspective and they follow the INVITE principle.
- To work in close collaboration with BA or maybe, wear the BA cap yourself (depends on the team dynamics)!!!!.
- To write Automated Acceptance Tests in BDD Format.
- To enforce Story Kick-offs.

We are the Quality Gate Keepers. We can ensure Quality at each layer by helping the teams to adapt BDD!!

PRESENTATION FINISHED
...ANY QUESTIONS?

THANKS FOR YOUR LISTENING...

Contact me for further queries here:

ruchichopra.in@gmail.com

@RuchiChopraTest

https://www.linkedin.com/in/ruchi-chopra-9ba07055