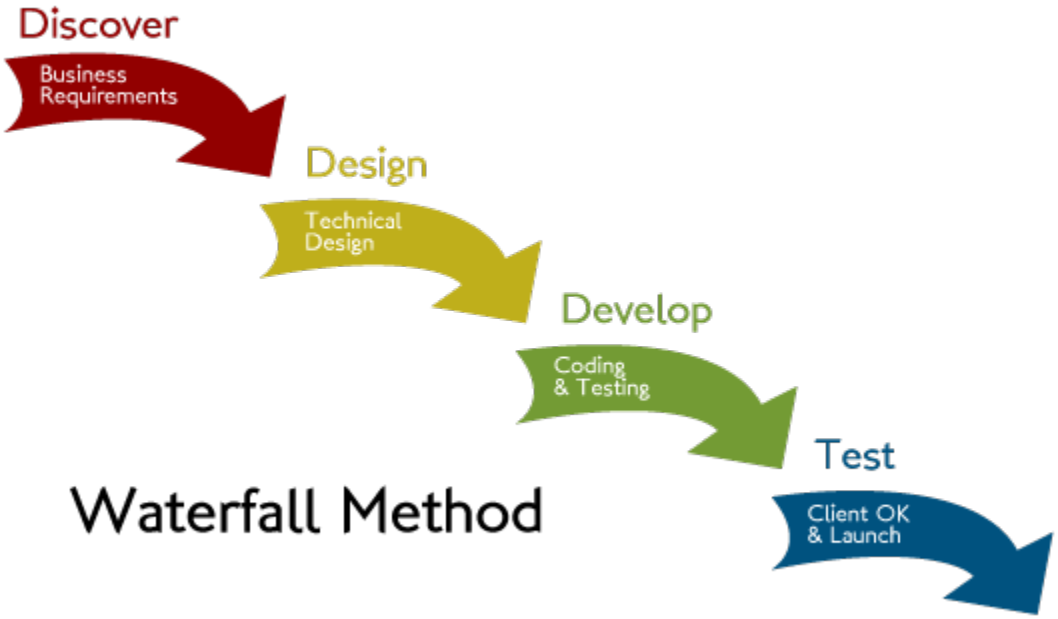
Behaviour Driven Development

Stuart Ashman & Marc Karbowiak

Ambiguous specifications?

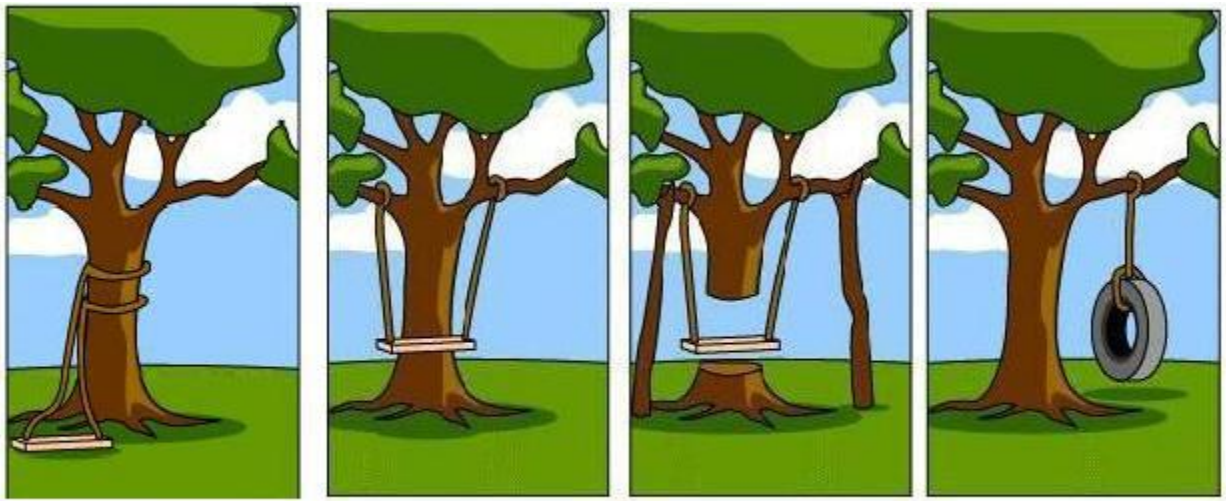
Most testing at the end?



Agile? Same language?

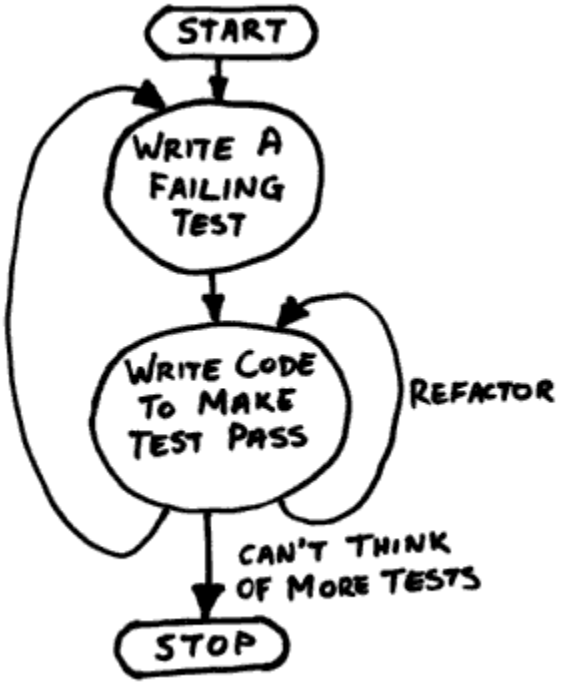
Deadlines approaching?

Get this kind of result?



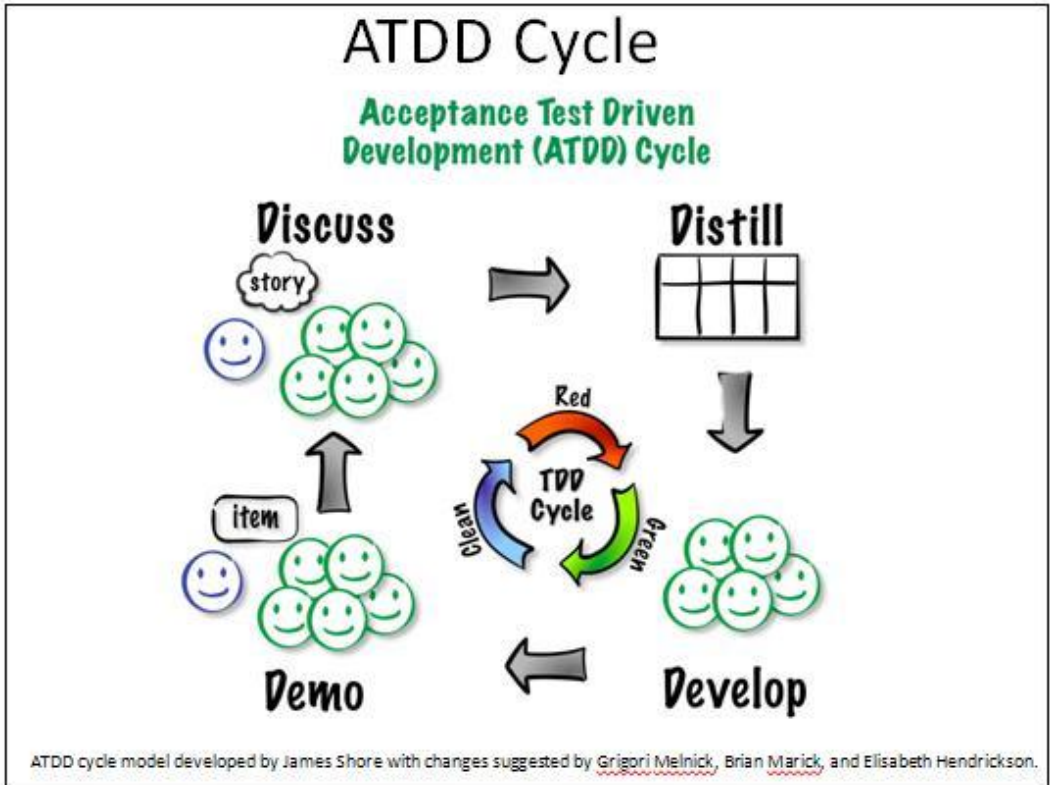
a) b) c) d)

So how can we do better?



+ Behaviours = BDD

Where do I start?



Discuss

Not just a development methodology, also a communications methodology

- 3 Amigos - PM/BA, Dev, QA + IxD

  in our case
- Common language
- Shared understanding

Distill

Scenario: Successful login attempt

Given I am on the login page
When I enter a valid email address
And a valid password
And I login
Then I will be presented with the product landing page

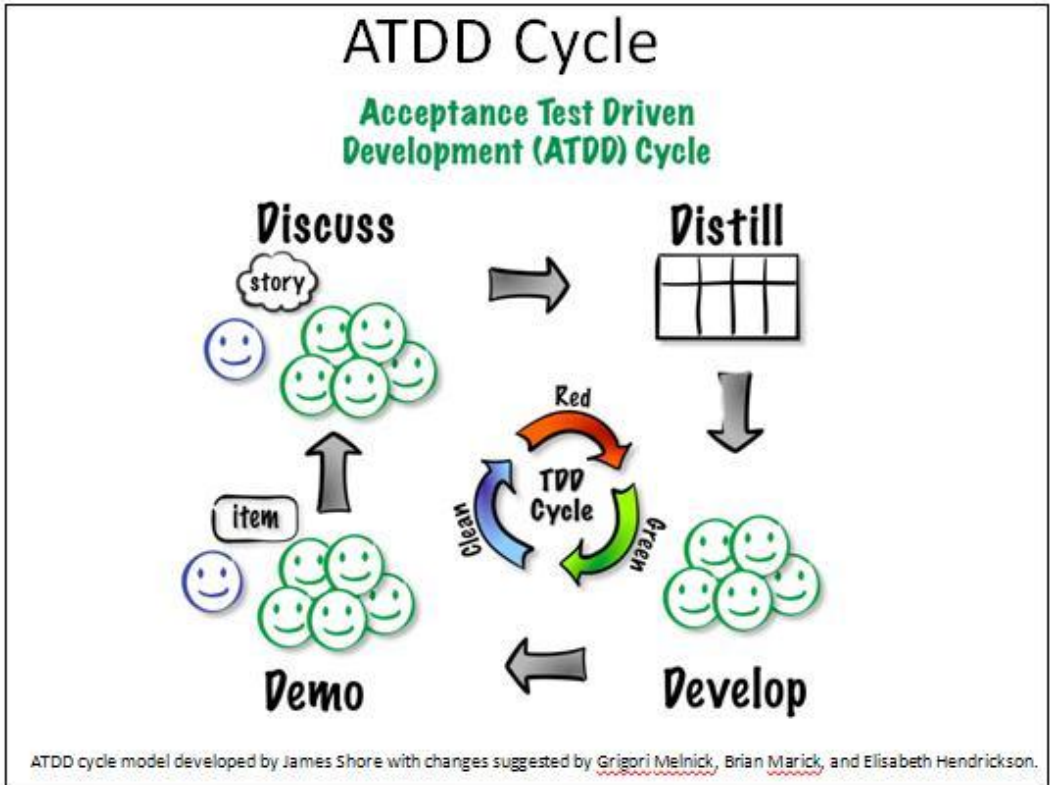
Develop

- Automate the behaviour specification
- Develop application code to pass test

Demo

- Demonstrate the working code using the automated tests
- Review your executable specifications
- Add tests to CI system
- Celebrate a job well done

Repeat - next part of story



What did I end up with?

- What the customer wanted
- Executable specifications
- Automated regression
- Kudos

What BDD is not

How to get started

Choose a tool to match your development language

Gherkin anyone?

Given - When - Then

Feature: Feedback when entering invalid credit card details

> In user testing we've seen a lot of people who made mistakes entering their credit card. We need to be as helpful as possible here to avoid losing users at this crucial stage of the transaction.

Background

Background:

Given I have chosen some items to buy

And I am about to enter my credit card details

Scenario

Scenario: Credit card number too short

When I enter a card number that's only 15 digits long
And all the other details are correct
And I submit the form
Then the form should be redisplayed
And I should see a message advising me of the correct number of digits

Excerpt From: Matt Wynne, Aslak Hellesøy. The Cucumber Book

A question of style

Declarative vs Imperative

- Avoid unimportant details
- Keep the scenario readable and meaningful
- Focus on the intent of the behaviour

Imperative Example

Scenario: Successful login attempt

Given I am on the login page
When I enter email as “example.user@mybiz.com”
And password as “Password1”
And I click the login button
Then the login form should be submitted
And I should see the product landing page with the title “Dashboard”

Declarative Example

Scenario: Successful login attempt

Given I am on the login page
When I enter a valid email address
And a valid password
And I login
Then I will be presented with the product landing page

DRY vs DAMP

DRY - Do not Repeat Yourself

DAMP - Descriptive And Meaningful Phrases

Try to tell a story rather than aiming for re-usability

Scenario Table Example

Scenario: Respond to survey
GIVEN A survey exists containing the following questions

Question	Question Type
What is your Age	Numeric
What is your Birthday	Date
What is your Name	String

WHEN I open the survey in responding

Scenario Table Example ...

THEN I should see the following questions

Question	Question Type
What is your Age	Numeric
What is your Birthday	Date
What is your Name	String

Scenario Outline Example

Scenario Outline: Successful login attempt

When I enter a valid email address <Email>
And a valid password <Pass>
And I login
Then I will be logged in as <User>
And I will be presented with the product landing page

Scenario Outline Example

Examples:

Email	Pass	User
validuser1@mybiz.com	Password1	validuser1
validuser2@mybiz2.com	Password2	validuser2

Hooks

Execute some code before or after

- Before and after test run
- Before and after Feature
- Before and after Scenario block
- Before and after Scenario
- Before and after Step

Tags

Use tags to organise or execute Features and Scenarios
Examples;

- @NightlyTest
- @Integration
- @ReportManager
- @SurveyResponding

Hooks support tag filtering, tags can be combined with OR

Questions