



TESTINGMIND

COMBINING DDD AND BDD TO BOOST ACCEPTANCE TESTING AND TESTING AUTOMATION

LEONARDO SOUZA MATTOS
#TAS19

Philadelphia
April 5th, 2019

About me



Technical Coach

IT Services
Principal Consultant

Developer

20+ years
In the industry

Soccer coach

Licensed for kids
U10 - U16

Adventurer

Meyers Briggs
abbreviated test



Have you heard this before?

“I tested it how you
told me to”

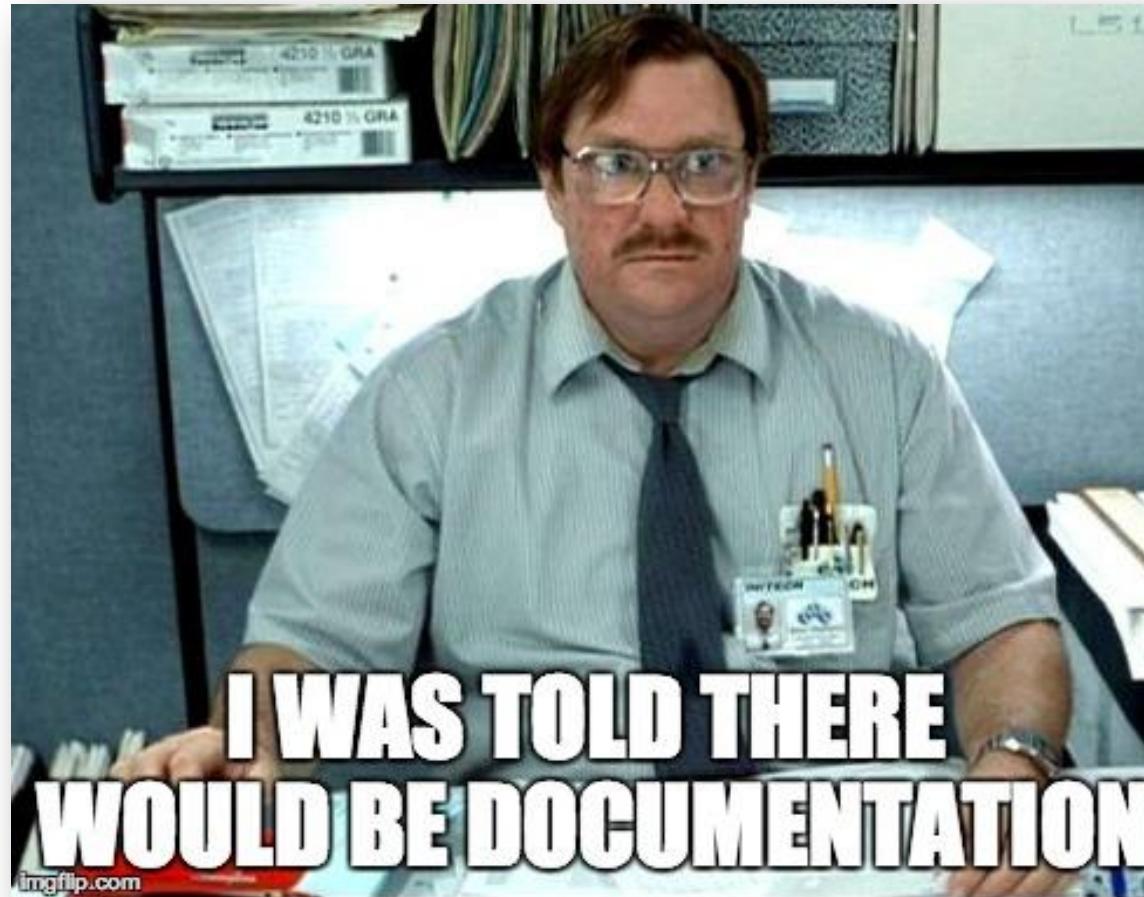
“Acceptance criteria
said this is expected”

“I assumed it was fine”

“When you say ‘account’, you
want to mean ‘client’, right?”

“I have not tested with this input. I
didn’t know it could happen”

“ I didn’t know what to test”



UP TO DATE DOCUMENTATION

**NOW THERE IS SOMETHING I
HAVEN'T SEEN IN A LONG TIME.**

```
73     return invTrans;
74 }
75
76 public void testBuyTransaction() {
77     InvestmentAccount invAccount = (InvestmentAccount) engine.getAccount( accountCode: "11311", currencySymbol: "EUR");
78     AssetAccount assetAccount = (AssetAccount) engine.getAccount( accountCode: "12111", currencySymbol: "EUR");
79     InvestmentTransaction invTrans = buildBuyTransaction(invAccount, assetAccount);
80
81     // test total amount of transaction
82     assertEquals(new BigDecimal(v: 1310.50).setScale(2), invTrans.getTotal().setScale(2));
83
84     // test market value of transaction from 3 January up to 8 January
85     assertEquals(new BigDecimal(i: 0).setScale(2), invTrans.getMarketValue(TD_minus_1).setScale(2));
86     assertEquals(new BigDecimal(v: 1030.50).setScale(2), invTrans.getMarketValue(TD).setScale(2));
87     assertEquals(new BigDecimal(v: 1029.50).setScale(2), invTrans.getMarketValue(TD_plus_1).setScale(2));
88     assertEquals(new BigDecimal(v: 1027.50).setScale(2), invTrans.getMarketValue(TD_plus_2).setScale(2));
89     assertEquals(new BigDecimal(v: 1311.50).setScale(2), invTrans.getMarketValue(SD).setScale(2));
90     assertEquals(new BigDecimal(v: 1332.50).setScale(2), invTrans.getMarketValue(SD_plus_1).setScale(2));
91
92     // test cash balance of transaction from 3 January up to 8 January
93     assertEquals(new BigDecimal(i: 0).setScale(2), invTrans.getAmount(assetAccount, TD_minus_1).setScale(2));
94     assertEquals(new BigDecimal(v: -1030.50).setScale(2), invTrans.getAmount(assetAccount, TD).setScale(2));
95     assertEquals(new BigDecimal(v: -1030.50).setScale(2), invTrans.getAmount(assetAccount, TD_plus_1).setScale(2));
96     assertEquals(new BigDecimal(v: -1030.50).setScale(2), invTrans.getAmount(assetAccount, TD_plus_2).setScale(2));
97     assertEquals(new BigDecimal(v: -1310.50).setScale(2), invTrans.getAmount(assetAccount, SD).setScale(2));
98     assertEquals(new BigDecimal(v: -1310.50).setScale(2), invTrans.getAmount(assetAccount, SD_plus_1).setScale(2));
99 }
100
101 public void testSellTransaction() {
102     InvestmentAccount invAccount = (InvestmentAccount) engine.getAccount( accountCode: "11311", currencySymbol: "EUR");
103     AssetAccount assetAccount = (AssetAccount) engine.getAccount( accountCode: "12111", currencySymbol: "EUR");
104     InvestmentTransaction invTrans = buildSellTransaction(invAccount, assetAccount);
105
106     // test total amount of transaction
107     assertEquals(new BigDecimal(v: 1310.50).setScale(2), invTrans.getTotal().setScale(2));
108 }
```

What was wrong with that?



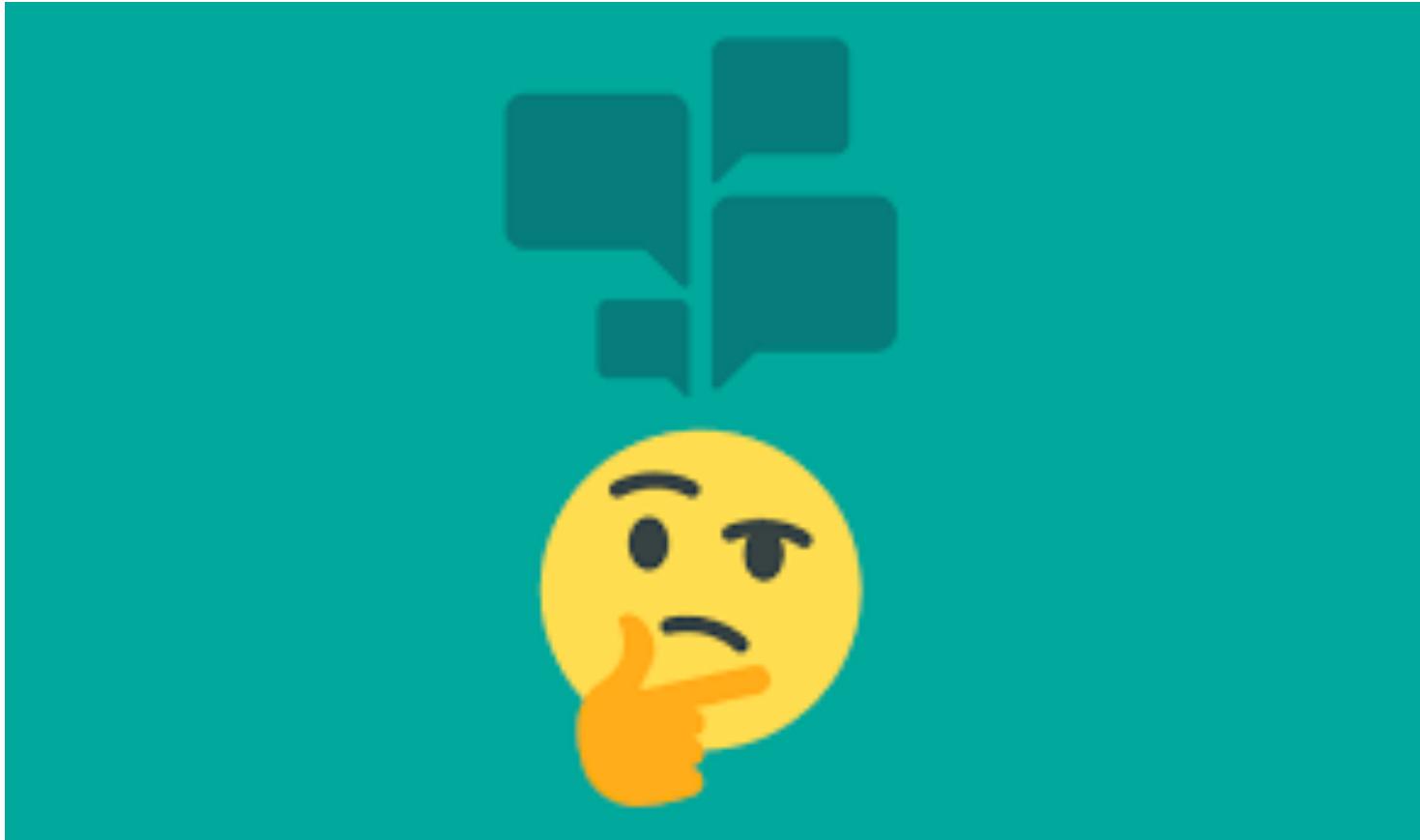
Tests weren't saying what the application does



Created by developers, potentially after the code was done



If fails doesn't tell me nothing. Need to go through the code to figure out something.



Is this
TDD?

What is
the
domain?

```
[code language="text"]
```

Scenario: Redirect user to originally requested page after logging in

Given a User "dave" exists with password "secret"
And I am not logged in

When I navigate to the home page
Then I am redirected to the login form

When I fill in "Username" with "dave"
And I fill in "Password" with "secret"
And I press "Login"
Then I should be on the home page

```
[/code]
```

What was
wrong with
that?



Misunderstood BDD : just
because its Gherkin does not
mean its BDD



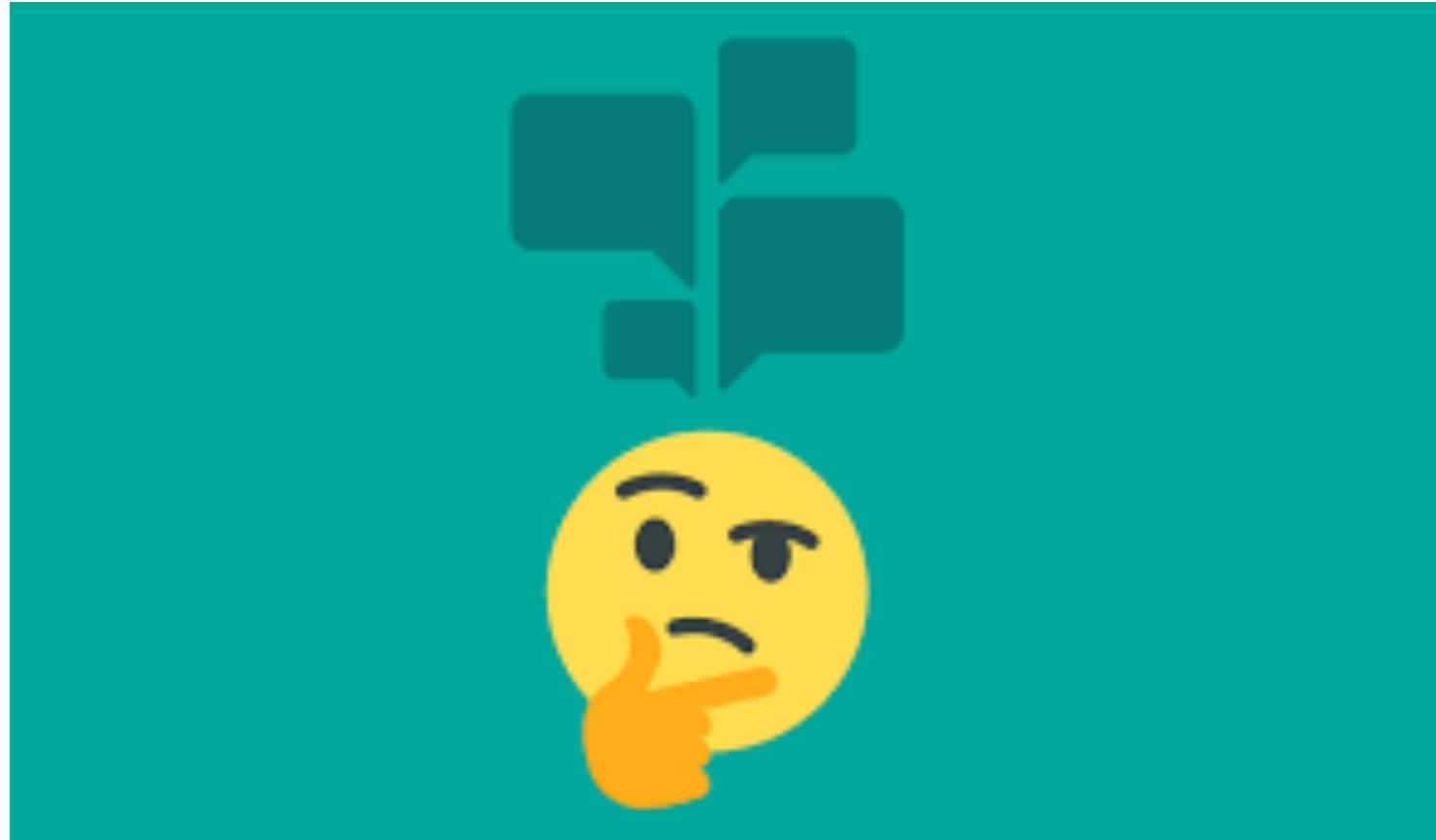
Design driven by the tool



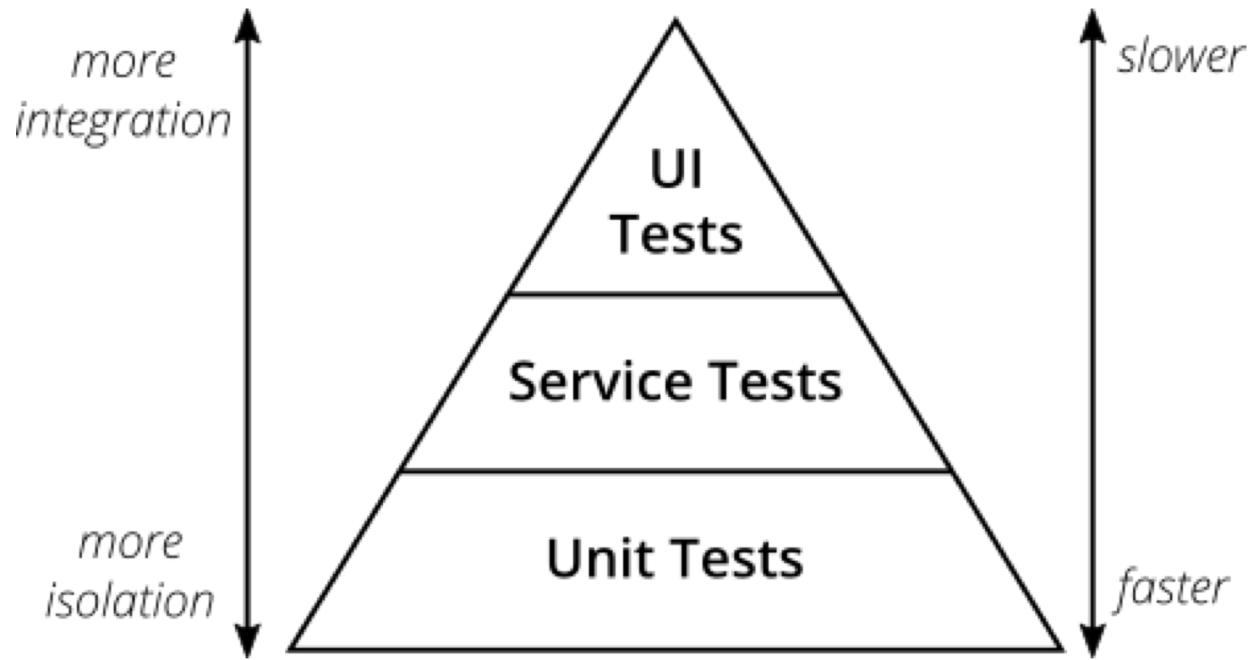
Use the “new thing” without
knowing how it really works.

What is the
story?

What is the
expected
behavior?



There are
many testing
pyramids



What is the
solution?



Domain language throughout tests, combining DDD and BDD

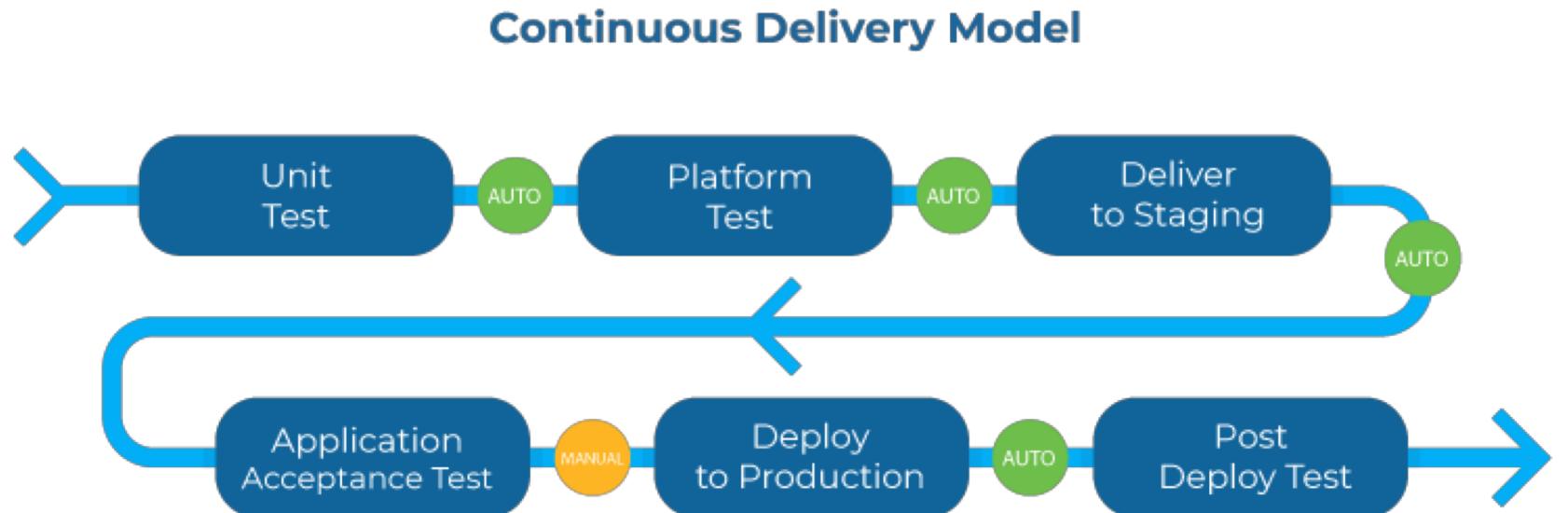
Its all about story:

DDD is about empathy. Its about knowing each other , knowing the model.

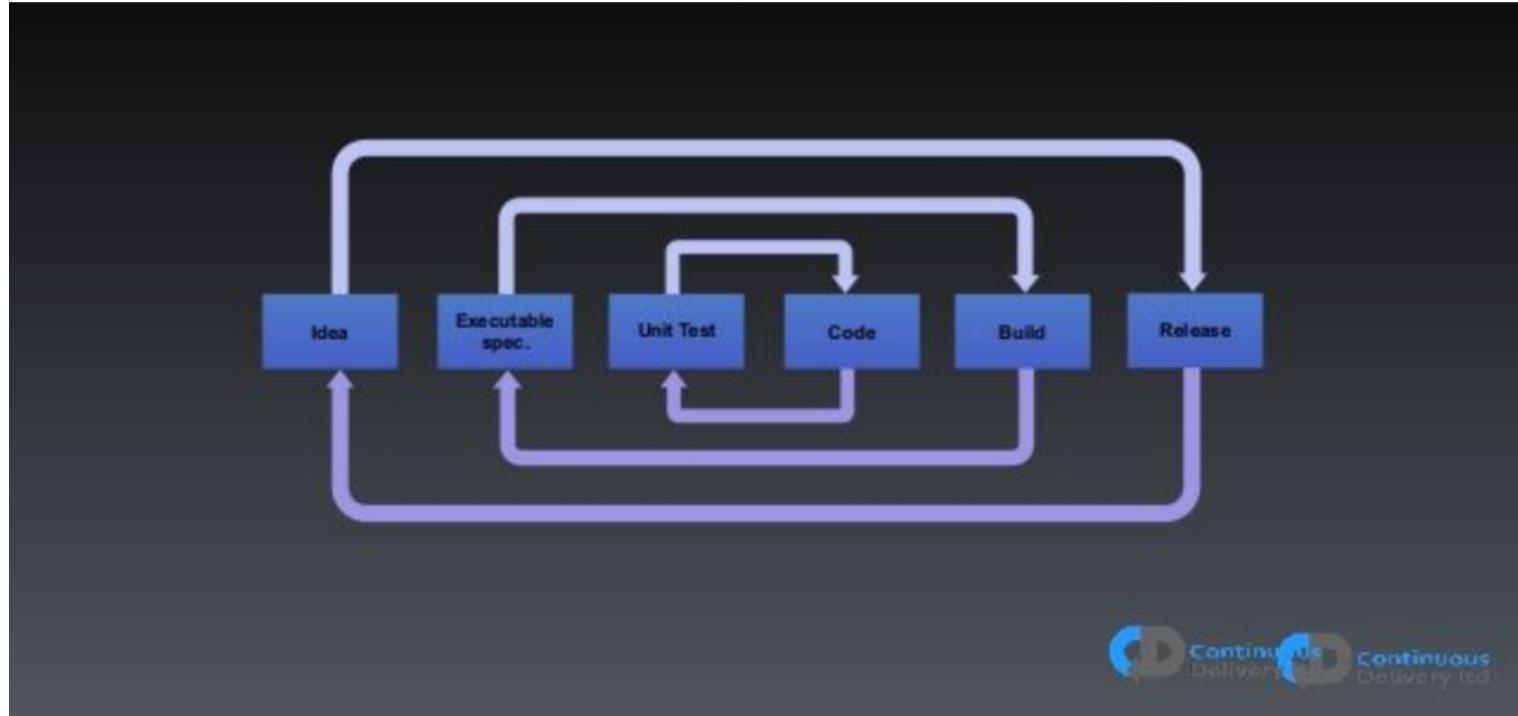
BDD is all about story and what is the story of what we are building.



Testing pyramids are not a good map



Feedback loops



Navigation by public transport

As a commuter

I want to know the fastest route between two places,

In order to get to my destination as quickly as possible

1 | **Feature:** Navigation by public transport

2 |

3 | As a commuter,

4 | I'd like to know the fastest route between two places,

5 | So that I get to my destination as quickly as possible

6 |

7 | **Scenario:** Navigate by public transport

8 |

9 | **Given** Intercity trains from Utrecht CS leave at 16:58, 17:08, 17:18

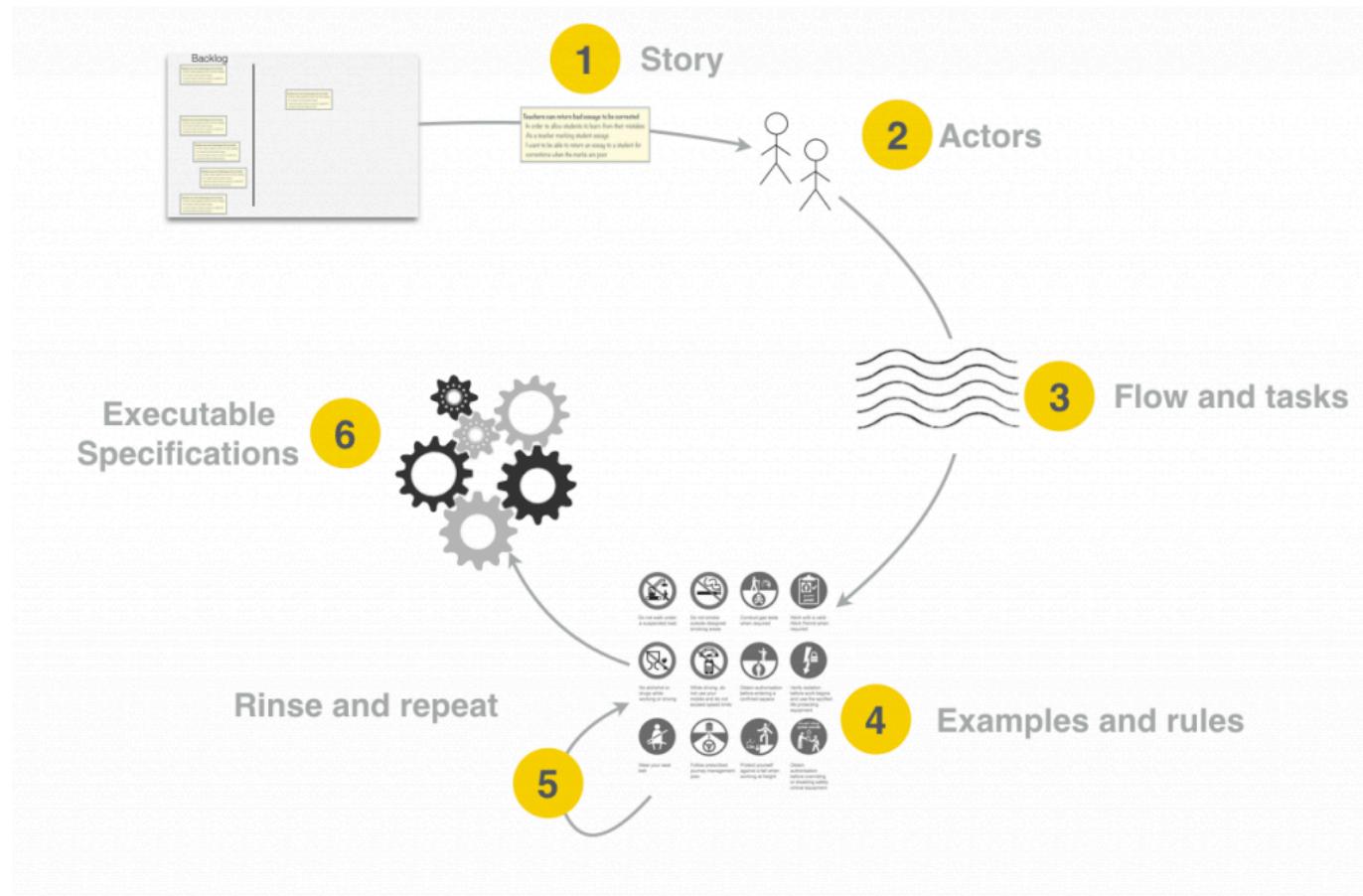
10 | **When** Connie wants to travel from Utrecht CS to Amsterdam CS at 17:00

11 | **Then** she should be told about the trains departing at 17:08, 17:18

12 |

DDD + BDD in practice:

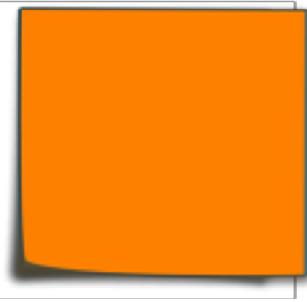
Feature Mapping!!



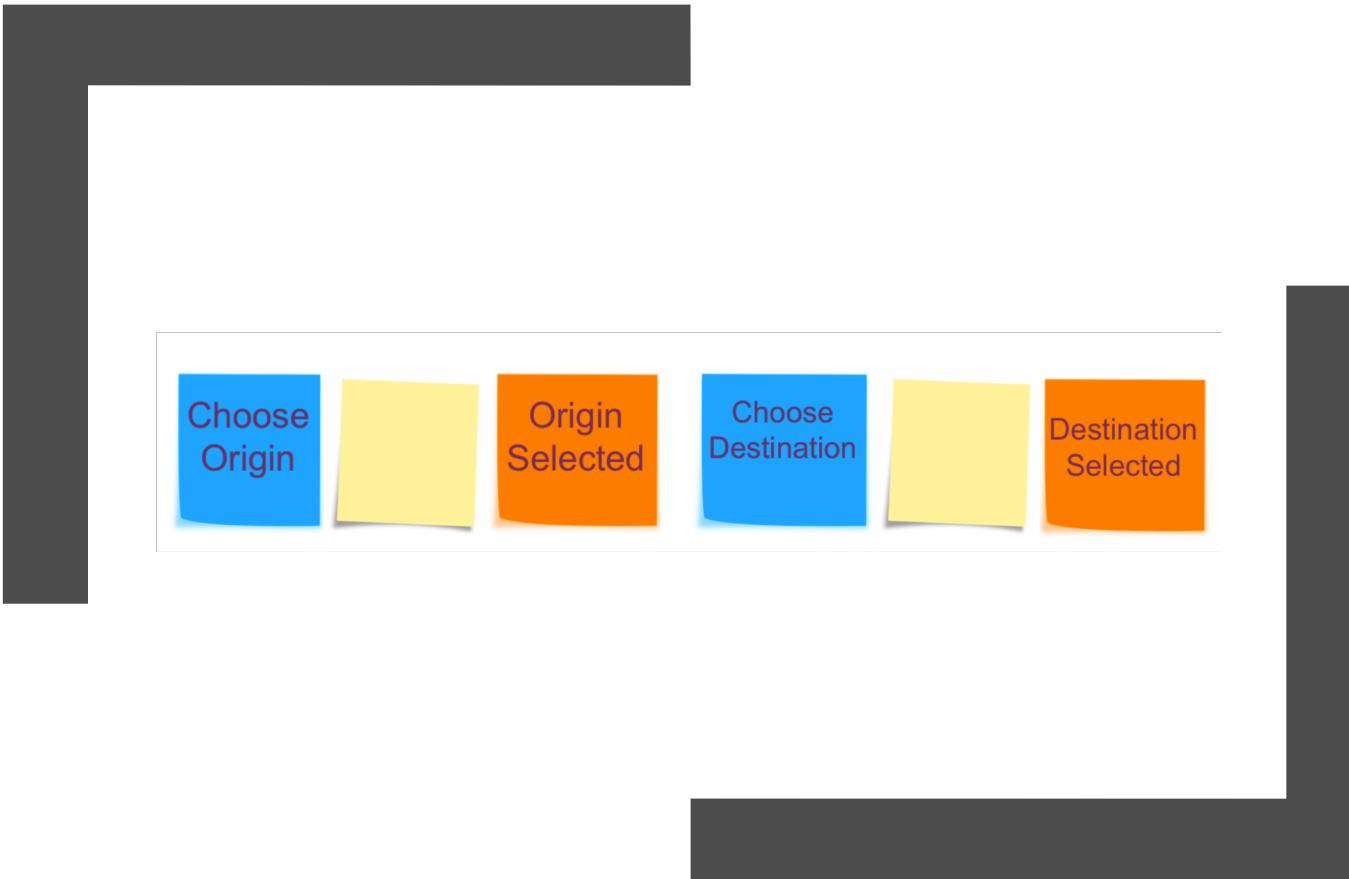


How feature mapping helps on acceptance testing

- This technique is all about creating specification based on examples, communicating in human language what the software does. It's a map on how is the flow of feature implementation
- Applying event storming and “Examples and Rules”, and making the “Spec by Example” embedded on your testing code



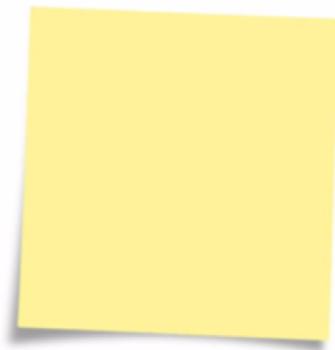
Event storming



Event Storming is a form of organizational anthropology; it's all about discussing the ***flow*** of events in your organization and modelling that flow in an easy to understand way.

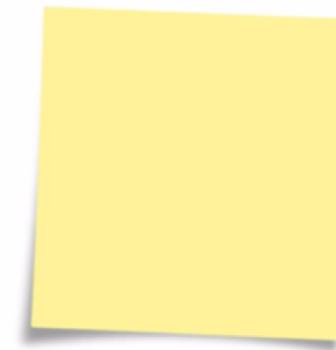
In our example

Choose
Origin



Origin
Selected

Choose
Destination



Destination
Selected

Example Mapping

This comes from Cucumber, from BDD

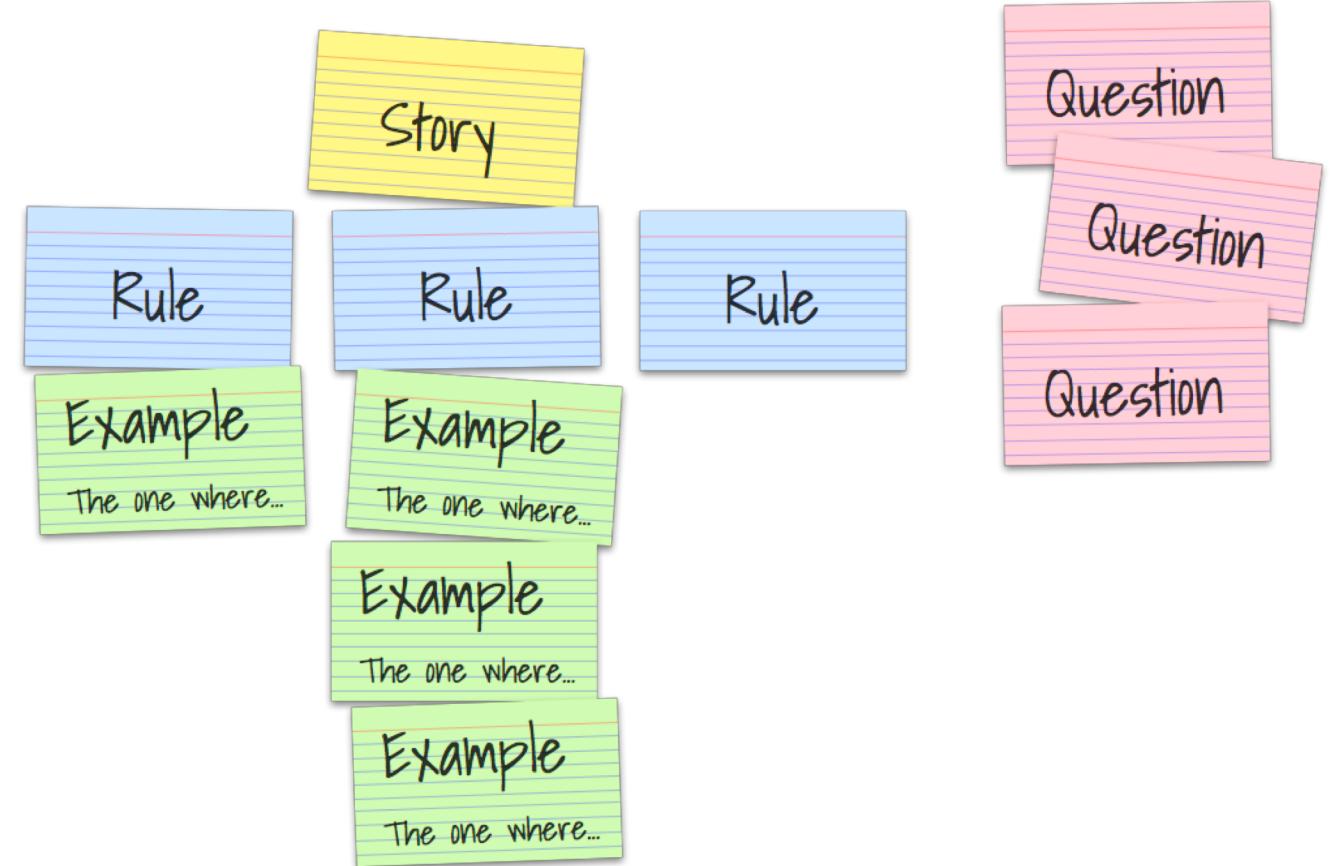
Discovery phase. Flipping out examples

Domain experts needs to be in the room

Get 4 coloured cards

Don't think initially on Gerkhin. Goes like "friends episode names" : "The one when"

If there is discussion or questions, you park on pink cards if cannot be solved in 15 min or no experts



Navigation by public transport

As a commuter

I want to know the fastest route between two places,

In order to get to my destination as quickly as possible

Acceptance Criteria

Given Ultrecht trains leave at 16:58, 17:08 and 17:18

When Connie wants to travel to Amsterdam at 17:00

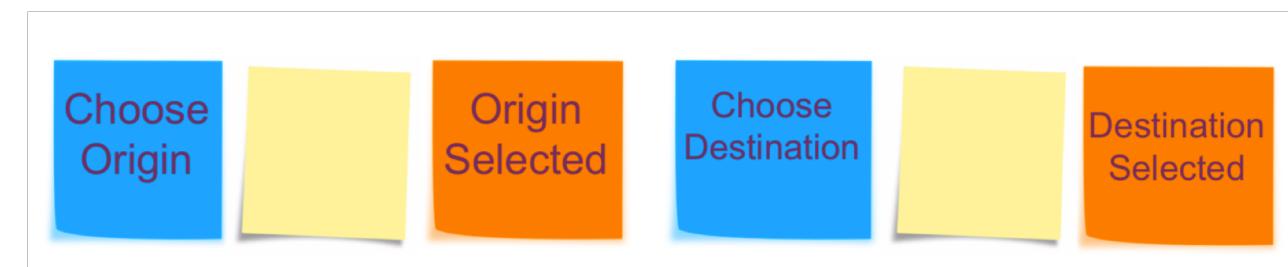
Then the available trains are 17:08 and 17:18

Given Ultrecht trains leave at 16:58, 17:08 and 17:18

When Connie wants to travel to Amsterdam at 18:00

Then no trains are available

And message “want to adjust the travel time to latest train?” shows up



journey_by_public_transport.feature

```

1 import {ChooseDestination, ChooseOrigin, PlanAJourney} from '../../../../../cucumber/support/saving_time/plan_a_journey';
2 import {JourneyList} from '../support/saving_time/journey_list';
3
4 const expect = global['chai'].expect;
5
6 const TRAIN_LIST_CSS = ".section-listbox";
7
8 describe('Navigation by public transport', () => {
9
10   before(callback => {
11     // Setup environment data and stuff.....
12   });
13
14   it('Navigate by public transport', () => {
15
16     PlanAJourney();
17     ChooseOrigin({ origin: 'Utrecht Centraal' });
18     ChooseDestination({ destination: 'Amsterdam Centraal' });
19
20     expect(JourneyList.Train_List.isPresent()).to.eventually.be.true;
21
22   });
23 })

```

```
7  
8 const FROM_INPUT_CSS = "#sb_ifc51 .tactile-searchbox-input";  
9 const TO_INPUT_CSS = "#sb_ifc52 .tactile-searchbox-input";  
10  
11 export function PlanAJourney() {  
12   browser.get( destination: '/maps' );  
13  
14   let navButtonElement = element(by.css(NAV_BUTTON_CSS));  
15   browser.wait(EC.visibilityOf(navButtonElement), 5000);  
16   navButtonElement.click();  
17  
18   let trainButtonElement = element(by.css(TRAIN_BUTTON_CSS));  
19   browser.wait(EC.visibilityOf(trainButtonElement), 5000);  
20   trainButtonElement.click();  
21 }  
22  
23 export function ChooseOrigin(origin: string) {  
24   let fromInputElement = element(by.css(FROM_INPUT_CSS));  
25   browser.wait(EC.visibilityOf(fromInputElement), 5000);  
26   return fromInputElement.sendKeys(origin);  
27 }  
28  
29 export function ChooseDestination(destination: string) {  
30   let toInputElement = element(by.css(TO_INPUT_CSS));  
31   browser.wait(EC.visibilityOf(toInputElement), 5000);  
32   toInputElement.sendKeys(destination);  
33   return toInputElement.sendKeys(protractor.Key.RETURN);  
34 }  
35
```

WebDriver

```
+import ...

const expect = global['chai'].expect;

const TRAIN_LIST_CSS = ".section-listbox";

export = function myStepDefinitionsWrapper() {

    this.Given(/^Intercity trains from Utrecht CS leave at 16:58, 17:08, 17:18$/, 
        // Setup environment data and stuff.....
        callback());
    });

    this.When(/^Connie wants to travel from Utrecht CS to Amsterdam CS at 17:00$/,
        PlanAJourney();
        ChooseOrigin( origin: 'Utrecht Centraal');
        return ChooseDestination( destination: 'Amsterdam Central');

    });

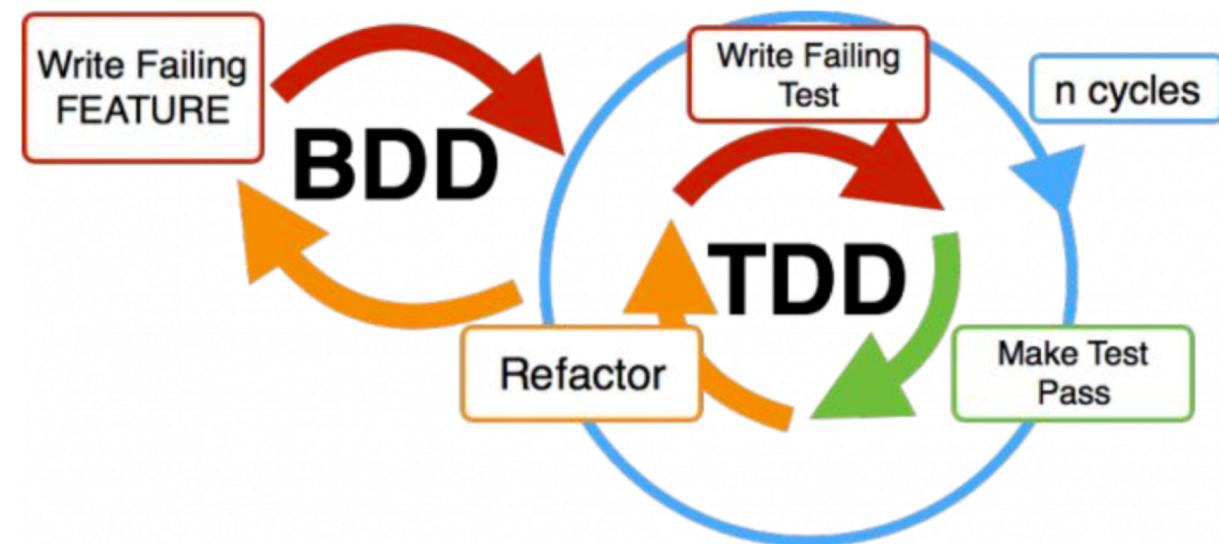
    this.Then(/^she should be told about the trains departing at 17:08, 17:18$/,
        return expect(JourneyList.Train_List.isPresent()).to.eventually.be.true;
    });

};
```

Cucumber

Conclusion

- Enhance your acceptance testing with executable spec by example (they can be written in Excel, Cucumber, Fitnesse, JSON files, etc...)
- Empathy from DDD + Story from BDD + Science = Better testing via quicker feedback (Your CD thanks you)
- It can and should be done from all test levels



Exit Criteria

How would be those concepts applied others testing types, like Unit Testing??

References

- [Adzic, Gojko](#) (2011). *Specification by example: How successful teams deliver the right software*. Manning. [ISBN 9781617290084](#).
- Pugh, Ken (2011). *Lean-Agile Acceptance Test Driven Development: Better Software Through Collaboration: A Tale of Lean-Agile Acceptance Test Driven Development*. Addison Wesley. [ISBN 978-0-321-71408-4](#).
- [Jump up to:^a ^b ^c](#) Larman, Craig; Vodde, Bas (2010). *Practices for Scaling Lean and Agile Development: Large, Multisite, and Offshore Product Development with Large-Scale Scrum*. Pearson. [ISBN 978-0-321-63640-9](#).
- [Jump up to:^a ^b](#) [Adzic, Gojko](#) (2009). *Bridging the Communication Gap: Specification by Example and Agile Acceptance Testing*. Neuri. [ISBN 0-9556836-1-0](#).
- [Crispin, Lisa](#); Gregory, Janet (2008). *Agile Testing: A Practical Guide for Testers and Agile Teams*. Addison Wesley. [ISBN 978-0-321-53446-0](#).
- *Pattern Languages of Program Design 2*. Addison-Wesley. 1996. [ISBN 978-0-201-89527-8](#).
- Ward Cunningham. ["EPISODES: A Pattern Language of Competitive Development Part I"](#). C2.com. Retrieved 2014-01-08.
- Martin Fowler 18 March 2004 (2004-03-18). ["SpecificationByExample"](#). Martinfowler.com. Retrieved 2014-01-08.
- [Beck, K.](#) (1999). *Extreme Programming Explained: Embrace Change*. Addison-Wesley. [ISBN 978-0-321-27865-4](#).
- [Evans, Eric](#) (2004). *Domain-Driven Design:Tackling Complexity in the Heart of Software*. Addison-Wesley. [ISBN 0-321-12521-5](#).
- [Weinberg, Gerald](#); Gause, Donald (1989). *Exploring Requirements: Quality Before Design*. Dorset House. [ISBN 0-932633-13-7](#).

