

Workshop: What Not To Test?



Alexandre Freire Kawakami
@freire_da_silva
alex@industriallogic.com



Copyright Industrial Logic 2017

Exercise 1

Application with many types of products. Devs new to TDD. Products have a complex algorithm to calculate discounts.

```
77 @Test
78 public void setPrice_() {
79
80     Item item = new BasicItem(Type.BOOK, "Fahrenheit 451", 0.0, true);
81     Double expected = 0.0;
82     Double actual = item.getPrice();
83     assertEquals(expected, actual);
84
85     //set new price
86     item.setPrice(1.0);
87     expected = 1.0;
88     actual = item.getPrice();
89     assertEquals(expected, actual);
90
91 }
92
93 @Test
94 public void isImported_true() {
95
96     Item item = new BasicItem(Type.BOOK, "Fahrenheit 451", 0.0, true);
97
98     assertTrue(item.isImported());
99
100 }
101
102 @Test
103 public void isImported_false() {
104
105     Item item = new BasicItem(Type.BOOK, "Fahrenheit 451", 0.0, false);
106
107     assertFalse(item.isImported());
108
109 }
110
Test: 111 }
25
26 @Override
27 public Type getType() {
28     return type;
29 }
30
31 @Override
32 public String getName() {
33     return name;
34 }
35
36 @Override
37 public Double getPrice() {
38     return price;
39 }
40
41 @Override
42 public void setPrice(Double pPrice) {
43     this.price = pPrice;
44 }
45
46 @Override
47 public boolean isImported() {
48     return imported;
49 }
50
Code: 51 }
```

Refactor

Re-write

Keep

Delete

Move

?

Exercise 2

Web App using Ruby on Rails framework that allows candidates for office to list and get feedback from constituents on their policy proposals.

Test:

```
1 require 'spec_helper'
2 describe Candidate do
3   context 'associations' do
4     it { should have_many(:proposals) }
5   end
6
7   context 'validations' do
8     it { should validate_presence_of :name }
9
10    it { should ensure_lenght_of(:phone).
11          is_at_least(7).
12          is_at_most(14)
13    }
14
15    it { should_not
16          allow_value('blah').for(:site) }
17
18    it { should
19          allow_value('http://www.blah.com')
20          .for(:site) }
21  end
22end
```

Code:

```
1 require 'valid_url'
2 class Candidate < ActiveRecord::Base
3   has_many :proposals
4
5   validates :name, presence: true
6
7   validates :phone, :length =>
8     {:in => 8..14},
9     :allow_blank
10
11  validates :site, :url => true,
12            :allow_nil => true
13
14end
```

Refactor

Re-write

Keep

Delete

Move

?

Exercise 3

Same Ruby on Rails Web App as exercise 2. The 'double' method gives us a mock of the object.

Test:

```
1 require 'spec_helper'
2 describe Candidates do
...
24 context 'search' do
25
26   before :each do
27     let(:candidate_mock) {double 'Candidate'}
28     candidate_mock.should_receive(:name).and_return('Joe')
29     Candidate
30     .should_receive(:find)
31     .with(1).and_return(candidate_mock)
32   end
33
34   it 'should find the right candidate' do
35     joe = Candidate.find(1)
36     joe.name.should eql 'Joe'
37     joe.should eql candidate_mock
38   end
39 end
```

Code:

Same as exercise 2

Refactor
 Re-write

Keep
 Delete

Move
 ?

Exercise 4

Still the same web app. All the tests are passing and our build is green, however the client has complained that when candidates insert proposals the proposal category is no longer being saved. After some research the only impacting change was made by a developer worried about application security, it's the **bold and underscored line below** it changes behavior so only named fields can be mass-assigned from the request params array .

Test:

- 1 Feature: Create proposal
- 2 As a candidate
- 3 I want to post my proposals
- 4 So that voters can evaluate them
- 5
- 6 Scenario:
- 7 Given I am logged in
- 8 And I am posting a proposal
- 9 When
- 10 I fill all fields of the proposal
- 11 Then
- 12 I should see a success message

Code:

```
1 class ProposalController < ApplicationController
2   def create
3     @proposal = Proposal.new(params[:proposal])
4     if @proposal.save
5       flash[:notice] = 'Proposal saved!'
6       redirect_to :action => 'list'
7     else
8       flash[:error] = 'Erros saving proposal!'
9       render :action => 'new'
10    end
11  end
12end
```

```
1 class Proposal < ActiveRecord::Base
2   has_many :proposals
3   attr_accessible :title, :description
4   ....
```

Refactor
 Re-write

Keep
 Delete

Move
 ?

Exercise 5

Web Application with a page where you can buy different plans for a product. The acceptance test was written using Cucumber+webdriver. Webdriver **does not** have built-in steps for `mouse_over` and `mouse_out`, our team is taking too long to implement this test because they want to implement those helpers hacking into webdriver's Javascript engine.

Test:

- 1 Scenario: Client sees tooltip for plan
- 2 Given
- 3 I select the 'light' plan
- 4 When
- 5 I mouse over 'tooltip'
- 6 Then
- 7 I should see 'tooltip' content
- 8 When
- 9 I mouse out 'tooltip'
- 10 Then
- 11 I should not see 'tooltip' content

Code:

```
7 <script src="http://code.jquery.com/jquery-19.1.js"></script>
...
52 $('<code>.contract</code>').tooltip({content: 'very fine print lawyer speak...'});
...
237 <span class='contract'>Light Plan</span>
```

Refactor
 Re-write

Keep
 Delete

Move
 ?

Exercise 6

A Web Application with a shopping cart that uses a service to authenticate users. The test mocks the login service and returns a VIP client. No one has ever seen this test fail.

Test:

```
20 @Before
21 public void initializeServices() {
22     loginService = mock(LoginService.class);
23     when(loginService.auth(anyString(), anyString()))
24         .thenReturn(vipUser);
25     fakeFinder = new ServiceFinder();
26     fake.setLoginService(loginService);
27     cart = new ShoppingCart(fakeFinder);
28 }
29
30 @Test
31 public void shouldAddProductAndCalculateTotal(){
32     cart.setUsername("john");
33     cart.setPassword("secret");
34     cart.addProduct(product1);
35     cart.addProduct(product2);
36     cart.calculateTotal();
37 }
```

Code:

```
73 public double calculateTotal() {
74     LoginService login = this.serviceFinder.getLoginService();
75     User user = login.auth(this.username, this.password);
76     double discount = 1.0;
77     if(user.isVip())
78         discount = 0.5;
79     double total = 0;
80     for(Product p : getProducts())
81         total += p.getPrice();
82     return total * discount;
83 }
```

Refactor
 Re-write

Keep
 Delete

Move
 ?

Exercise 7

A movie script editor desktop application, when you left-click on a view icons are created in the relevant position to the left of the text. When you mouse over the icon a tooltip appears. The test uses SWTBot (SWT is a GUI library) to simulate mouse interaction with the app. The test passed in the developers' machine but broke in the Continuous Integration server.

Test:

```
116 @Test
117 public void shouldManipulateStructureMarks() throws Exception {
118     bot.leftClickAt(scenesView, 800, 508);
119     addStructureMarkAtLevel("Dramatic Unit", 1);
120
121     bot.leftClickAt(scenesView, 900, 508);
122     addStructureMarkAtLevel("Sequence", 2);
123
124     addStructureMarkAtLevel("Act", 3);
125
126     bot.verifyTooltipAtWithValue(30, 190, "ACT 2\nSEQ 2\nDU 3");
127
128     deleteAct();
129 }
130
131 private void deleteAct() {
132     int actCount = countStructureMarksAtLevel(3);
133     int sequenceCount = countStructureMarksAtLevel(2);
134
135     Widget editor = bot.activeEditor().getWidget();
136     bot.leftClickAt(editor, 30, 190);
137     bot.leftClickAt(editor, 170, 220);
138
139     assertThat(countStructureMarksAtLevel(2), is(sequenceCount));
140     assertThat(countStructureMarksAtLevel(3), is(actCount - 1));
141 }
142
```

The code is complicated and irrelevant for this exercise.

Refactor
 Re-write

Keep
 Delete

Move
 ?

Exercise 8

Web application that integrates with a mail webservice to calculate shipping. The integration test below worked well. After a year of having this project in production the client asked for a few changes, when the developers ran the tests this one test failed saying that it expected 8.2 but got 9.7.

Test:

```
1 require 'spec_helper'
2 describe AddressController do
3
4   it 'should calculate shipping' do
5     get :shipping, :zipcode => '90210'
6     assigns(:shipping).should == '8.2'
7   end
8
9 end
```

Code: It's really simple, it calls the webservice and parses the json result to extract the price.

Refactor

Re-write

Keep

Delete

Move

?

Exercise 9

Application that sends a confirmation e-mail to clients that placed a product order

Test:

```
public class InvoiceBuilderTest extends TestCase...
    public void testNotSureExactly() {
        Account account = new Account();
        account.createWith("Mike (Collaborations) Hill", "123 MyStreet", "", "Afto
        account.addCreditCard("Visa", "12345678910111213", "000", "01/2010", true)
        Accounts.initialize()
        Accounts.put(account);

        Order order = new Order(account,now());
        order.add(new Product("SKU100010", "Silly Putty", true, false));
        order.add(new Product("SKU100101", "Non-Line Item", true, true));
        Product master = new Product("SKU101010", "Master-Item", Dollar.zero(), Weigh
        order.add(master);

        Products.initializeDatabase();
        Products.add(new ProductDescriptor("SKU100010", "Silly Putty", new Dollar(0.
        Products.add(new ProductDescriptor("SKU100101", "Non-Line Item", new Dollar
        ProductDescriptor master2 = new ProductDescriptor("SKU101010", "Master-Item
        ProductDescriptor sub = new Product("SKU111111", "Sub-Item", new Dollar(1.00
        master2.addChild(sub);
        Products.add(master2);

        CardReader.setValidateOff();
        restartServer();
        copyScriptingFiles();
        openInternetExplorer();
        Session s = login("collaboratorDummy", "xyzy");
        stuffOrderFields(order, s);
        clickSubmit();
        Outlook o = new Outlook();
        o.restart();
        sleep(30000);
        String text = o.fetchLineContaining("Total:");
        String line = parseToTotalLine(text);
        assertEquals(new Dollar(2), Dollar.fromString(line.substring(line.size()-5)
    }
```

Code: they lost it, all we have is a jar.

Refactor
 Re-write

Keep
 Delete

Move
 ?

Exercise 10

In a 3 amigos meeting the team looked through this scenario the PM had written for a new feature.

One of the developers, after asking a few clarification questions, said: "Oh, if that's what you really want then we don't need to build anything! The user can get those localized documents if the admin enables customized panels and adds a filtered document panel with a i18n filter!"

Test:

Scenario:

Global business HR administrator makes localized documents available for his employees

Given I login as an admin

And I upload policy document 'Safety Training'

And I mark that document as specific for the EU region

And I upload policy document 'Sexual Harassment'

And I mark that document as specific for USA region

When I login as USA employee

Then I should see 'Sexual Harassment' document in my home page

When I login as Germany employee

Then I should see 'Safety Training' document in my home page

Code: they didn't write any, they also did not automate this scenario.

Refactor

Re-write

Keep

Delete

Move

?