



Domain Specific Languages for Selenium tests

Emily Bache, jfokus 2010

What is selenium?



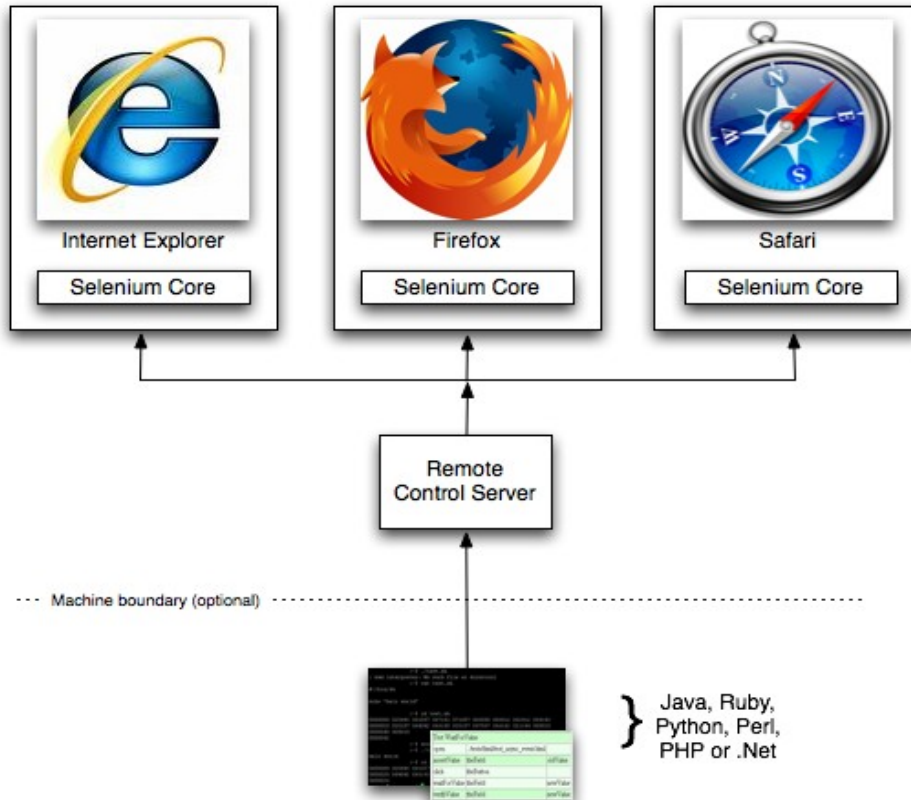
- Selenium is a suite of tools to automate web application testing
- Includes Selenium RC (Remote Control) & Selenium IDE





Selenium RC

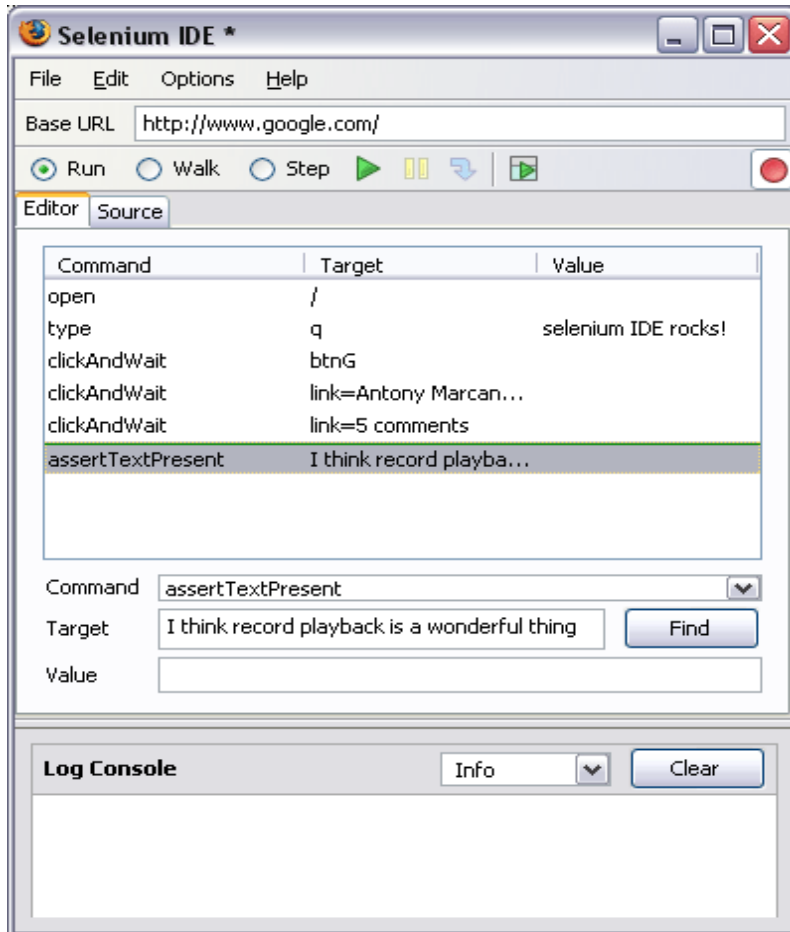
Windows, Linux, or Mac (as appropriate)...



- Write test in Java (etc)
- Run your web app in a browser



Selenium IDE



- Firefox plug-in for recording tests

Why choose selenium over other tools?

- Run tests in a real browser
 - Use real browser's javascript interpreter
- Much of selenium itself is written in Java
- Comparatively large user base
- Actively maintained – recent alpha release of version 2.0



Selenium 2.0

- new WebDriver API
- DefaultSelenium API unchanged from 1.0
- “bridge” class to ease migration.



Demo: Ad Orchestrator

Media Agency

Create Ad Campaign



Consumer

Sign Up



Publisher

Receive Ads

Provide Ad inventory





Screencast: Create Publisher

- [url of screencast demo](#)



@Test

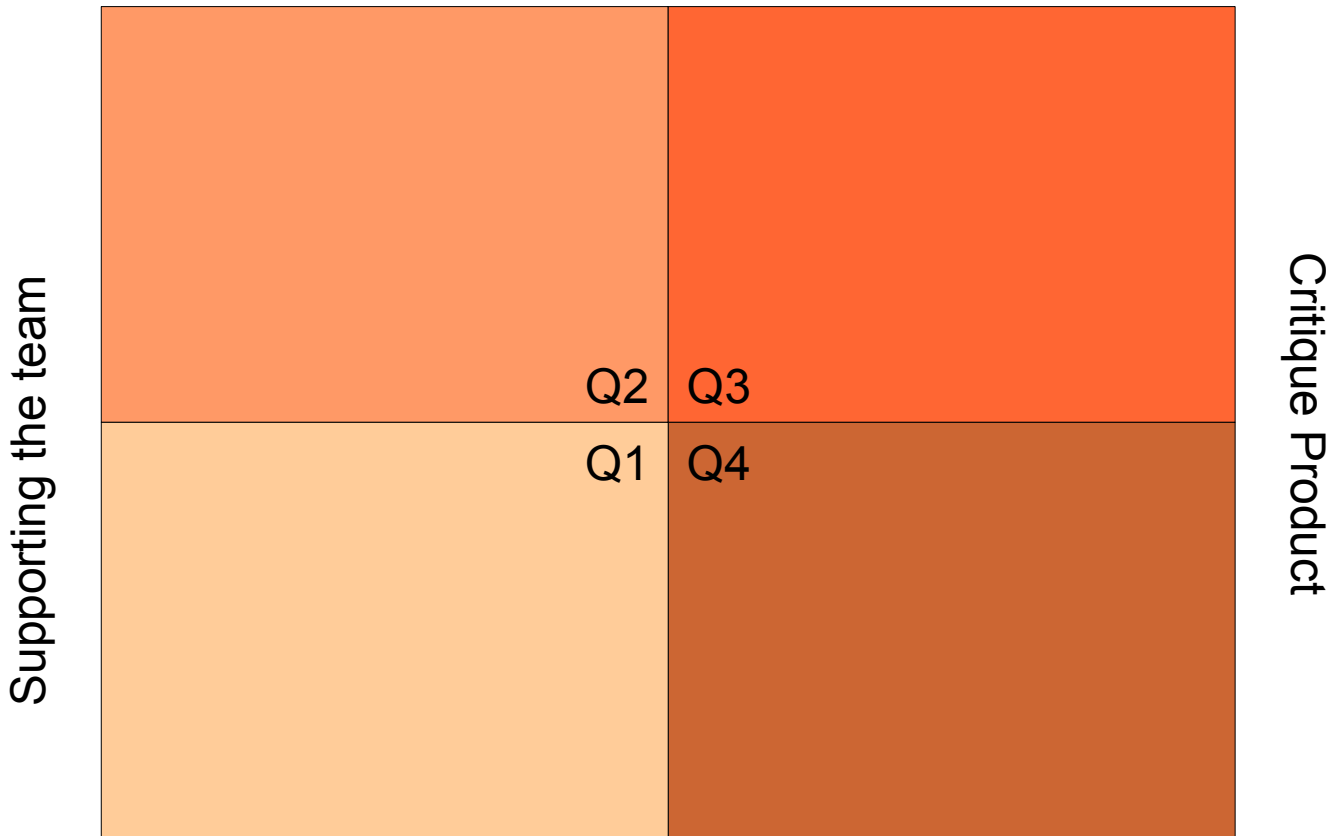
```
public void createNewPublisher() {  
    selenium.click("menuItemPNewPublisher");  
    selenium.waitForPageToLoad("30000");  
    selenium.type("name", "Disney");  
    selenium.type("address", "24 Stars Ave");  
    selenium.type("zipCode", "91210");  
    selenium.type("city", "Hollywood");  
    selenium.type("country", "USA");  
    selenium.type("contactPersonName", "Emily");  
    selenium.type("contactPersonEmail", "emily@drutt.com");  
    selenium.type("contactPersonPassword", "123456");  
    selenium.type("contactPersonRetypePassword", "123456");  
    selenium.click("save");  
    selenium.waitForPageToLoad("30000");  
  
    assertEquals(selenium.getTable("Users.1.1"), "Emily");  
}
```

Wait a minute, why are we doing this?

- Agile Tests:
 - support developers
 - critique the product

Agile Testing Quadrants

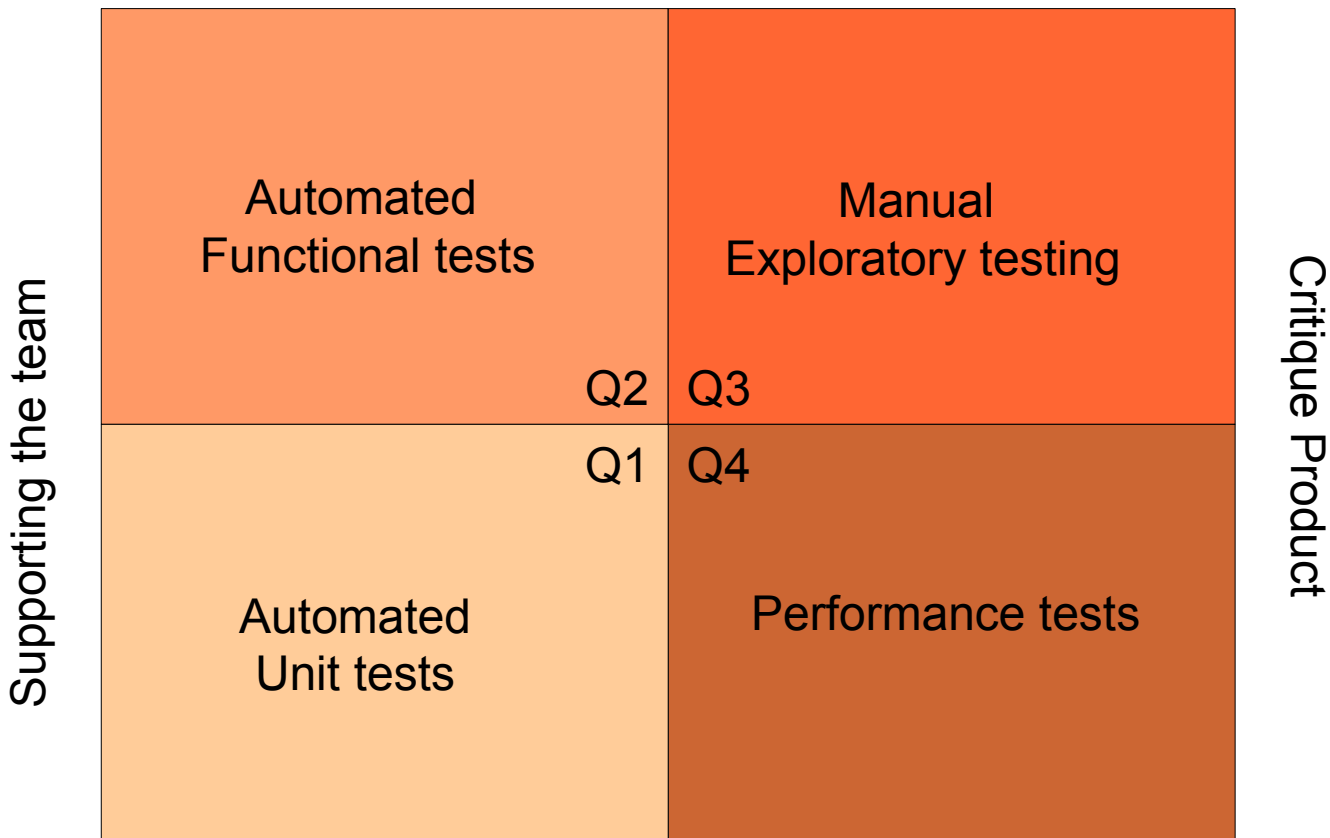
Business Facing



Technology Facing

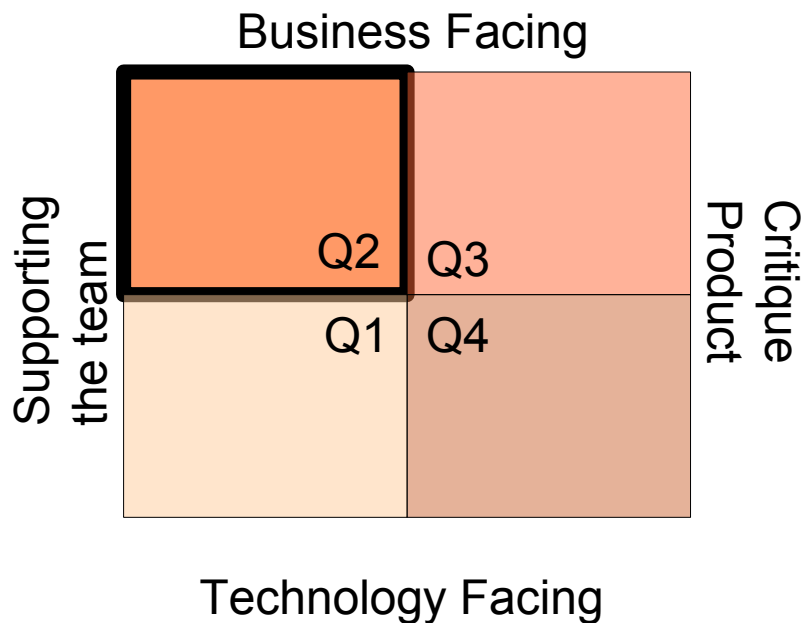
Agile Testing Quadrants - examples

Business Facing

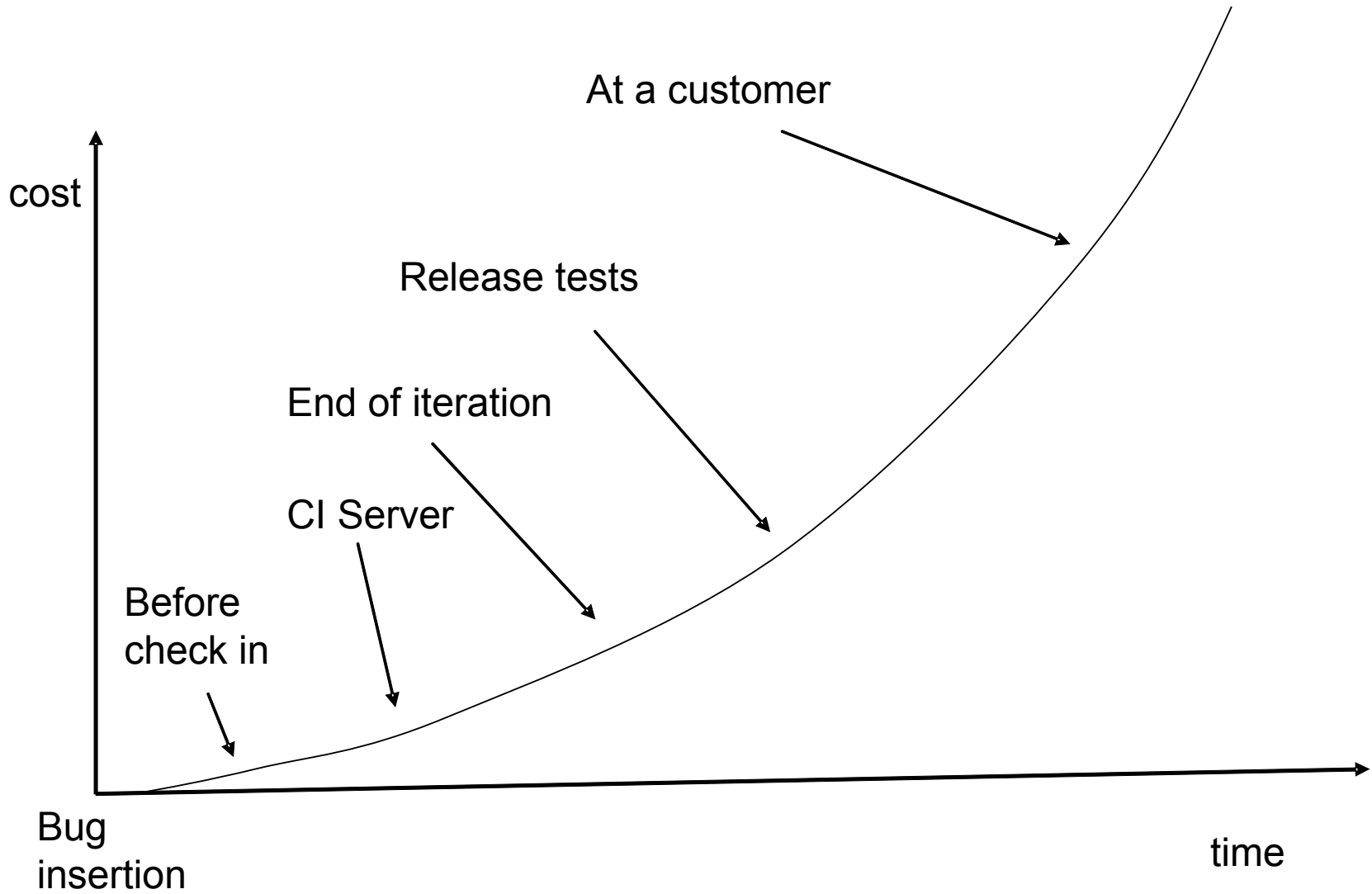


Technology Facing

Selenium tests are usually Q2



- Help developers understand the app from the user's point of view
- Identify regression bugs



Good tests have these properties:



Readability



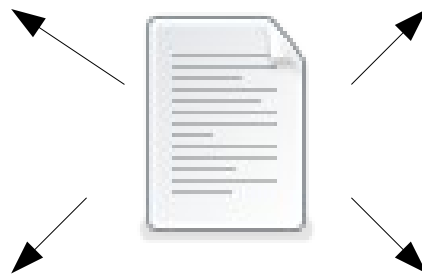
Robustness



Speed



Coverage





So Selenium tests should:

- Be run often, every build, every checkin, every day
 - *But they are often rather slow to execute*
- Clearly explain what features the app has
 - *But the IDE produces low abstraction widget-oriented script*
- Support refactoring
 - *But the tests are often rather fragile, eg xpaths break when the html changes*



Prefer ids to xpaths



Readability

```
selenium.selectFrame("mainFrame");  
selenium.click("//a[@id='menuItemPNewPublisher']/div");
```



Robustness

```
selenium.click("menuItemPNewPublisher");
```



Avoid sleep(), prefer waitFor()

```
try {  
    Thread.sleep(3000);  
} catch (InterruptedException e) {  
}
```



Robustness

```
selenium.waitForCondition(  
    "selenium.getValue(\"id='userTableStatus'\") == \"loaded\";", "3000");
```

OR:

```
public static void waitForElementForSeconds(Selenium selenium, String elementId,  
int seconds) {  
    for (int halfSecond = 0; halfSecond < seconds * 2; halfSecond++) {  
        if (selenium.isElementPresent(elementId)) {  
            return;  
        }  
        Util.sleep(500);  
    }  
    throw new RuntimeException("Time out - Element not found: " + elementId);  
}
```



Speed



Check for correct page and absence of error messages



Readability

```
assertTrue(selenium.isElementPresent("loginButton"),  
           "Not on login page");
```

```
//... do login ...
```

```
assertFalse(selenium.isVisible("badPassword"),  
            "Failed to log in: bad password");  
assertTrue(selenium.isElementPresent("logoutLink"),  
            "Failed to log in: no logout link shown");
```



Coverage



In @Before, remove alerts and frame selections

@Before

```
public void clearAlertsAndFrameSelections() {  
    while (selenium.isAlertPresent()) {  
        selenium.getAlert();  
    }  
    selenium.selectFrame("relative=top");  
    while (selenium.isAlertPresent()) {  
        selenium.getAlert();  
    }  
}
```



Robustness



Selenium tests are slow

- Develop tests in Firefox
- Use TestNG instead of JUnit
- Choose a representative subset to run at each checkin, < 10 mins to run
- Use Selenium Grid to parallelize
- Use WebDriver instead of DefaultSelenium



Speed



Selenium 2.0 : WebDriver vs DefaultSelenium

- **WebDriver**
 - is faster to execute tests
 - has clean API and support for PageObject pattern
 - has immature support for Selenium IDE and Grid
- **DefaultSelenium**
 - runs in any browser (almost)
 - has “kitchen sink” API



Organize your test code

- Data structs for form data
- PageObjects OR Scenario classes for domain actions



Readability



Robustness



Coverage



Data structs for form data



Readability

```
PublisherData newPublisher = new PublisherData("Disney")  
    .withAddress(  
        new AddressData("24 Stars Ave",  
                        "91210", "Hollywood", "USA"))  
    .withContactPerson(  
        new UserData("Emily", "emily@test.com",  
                    "123456"));
```



Robustness



Data structs for form data (2)



Readability

```
public class PublisherData {
```

```
    public String name;  
    public AddressData address;  
    public UserData user;
```

```
    public PublisherData(String name) {  
        this.name = name;  
    }
```

```
    public PublisherData withAddress(AddressData addressData) {  
        this.address = addressData;  
        return this;  
    }
```



Robustness



PageObject pattern

- The page structure of your app helps structure your test code
- WebDriver (Selenium 2.0) has built-in support for it



Readability



Robustness



Coverage



PageObject example

```
public class NewPublisherPage implements Page {
```

```
    private Selenium selenium;
```

```
    public NewPublisherPage(Selenium selenium) {  
        this.selenium = selenium;  
        selenium.click("menuItemPNewPublisher");  
        selenium.waitForPageToLoad("30000");  
    }
```

```
//...
```



Readability



Robustness



Coverage



PageObject example (2)



Readability



Robustness



Coverage

```
public class NewPublisherPage implements Page {  
    //...
```

```
    public void enterPublisherDetails(PublisherData pd) {  
        selenium.type("name", pd.name);  
        selenium.type("address", pd.address.firstLine);  
        selenium.type("zipCode", pd.address.zipCode);  
        selenium.type("city", pd.address.city);  
        selenium.type("country", pd.address.country);  
        selenium.type("contactPersonName", pd.user.name);  
        selenium.type("contactPersonEmail", pd.user.email);  
        selenium.type("contactPersonPassword",  
                    pd.user.password);  
        selenium.type("contactPersonRetypePassword",  
                    pd.user.password);  
    }  
    //...
```



PageObject example (3)



Readability

```
public class NewPublisherPage implements Page {
```

```
//...
```

```
public Page save() {  
    selenium.click("save");  
    selenium.waitForPageToLoad("30000");  
    if (selenium.isVisible("errors"))  
        return this;  
    return new ViewPublisherPage(selenium);  
}
```

```
//...
```



Robustness



Coverage



Test code with PageObject



TestCase

Controls what happens:

- which pages
- which data
- what to assert



Page Object

Encapsulate actions possible on a page.



Data Struct

Encapsulate form data



What the test looked like last time

@Test

```
public void createNewPublisher() {  
    selenium.click("menuItemPNewPublisher");  
    selenium.waitForPageToLoad("30000");  
    selenium.type("name", "Disney");  
    selenium.type("address", "24 Stars Ave");  
    selenium.type("zipCode", "91210");  
    selenium.type("city", "Hollywood");  
    selenium.type("country", "USA");  
    selenium.type("contactPersonName", "Emily");  
    selenium.type("contactPersonEmail", "emily@test.com");  
    selenium.type("contactPersonPassword", "123456");  
    selenium.type("contactPersonRetypePassword", "123456");  
    selenium.click("save");  
    selenium.waitForPageToLoad("30000");  
  
    verifyEquals(selenium.getTable("Users.1.1"), "Emily");  
}
```



CreatePublisher test with Page Object

@Test

```
public void createNewPublisher() {  
    PublisherData newPublisher = new PublisherData("Disney")  
        .withAddress(  
            new AddressData("24 Stars Ave",  
                "91210", "Hollywood", "USA"))  
        .withContactPerson(  
            new UserData("Emily", "emily.bache2@drutt.com",  
                "123456"));  
  
    NewPublisherPage page = new NewPublisherPage(selenium);  
    page.enterPublisherDetails(newPublisher);  
    ViewPublisherPage viewPage = (ViewPublisherPage)page.save();  
    viewPage.verifyPublisherDetails(newPublisher);  
}
```




Scenario classes for domain actions



Readability

```
public class PublisherScenarios {  
    Selenium selenium;
```

```
    public PublisherScenarios(Selenium selenium) {  
        this.selenium = selenium;  
    }  
}
```



Robustness

```
    public void createPublisher(PublisherData pd) {  
        openNewPublisherPage();  
        enterPublisherDetails();  
        save();  
    }  
}
```

```
    public void openNewPublisherPage() {  
        selenium.click("menuItemNewPublisher");  
        selenium.waitForPageToLoad("30000");  
    }  
    //...
```



Refactored test with Scenario and Data Structs

@Test

```
public void createNewPublisher() {  
    PublisherData newPublisher = new PublisherData("Disney")  
        .withAddress(  
            new AddressData("24 Stars Ave",  
                "91210", "Hollywood", "USA"))  
        .withContactPerson(  
            new UserData("Emily", "emily@test.com",  
                "123456")));  
  
    PublisherScenarios scenarios =  
        new PublisherScenarios(selenium);  
    scenarios.createPublisher(newPublisher);  
  
    scenarios.verifyPublisherDetails(newPublisher);  
}
```



Test code in layers



TestCase

Controls what happens:

- which actions
- which data
- what to assert



Scenarios

Methods for domain actions.



Data Struct

Encapsulates form data.

Demo: Ad Orchestrator

Media Agency

Create Ad Campaign



Consumer

Sign Up



Publisher

Receive Ads

Provide Ad inventory



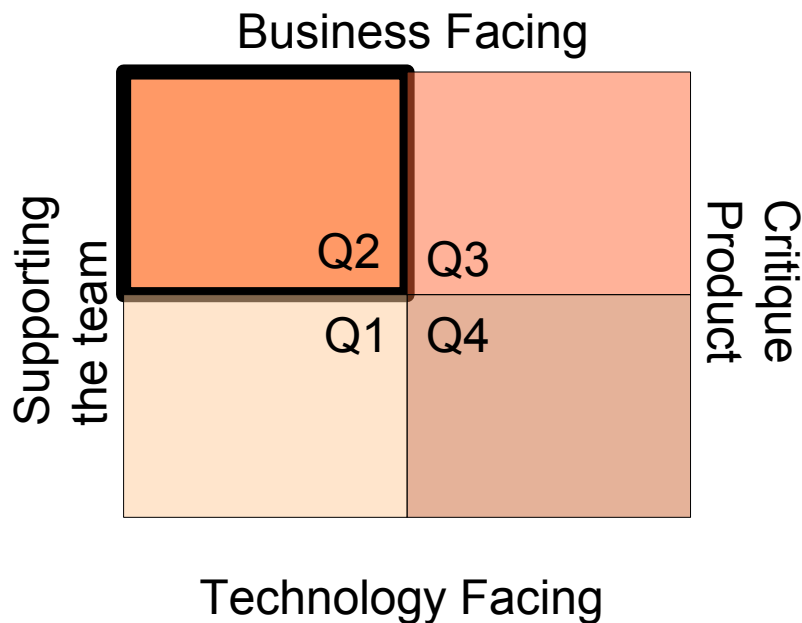


Testing Business Rules in Ad Orchestrator

@Test

```
public void testGenderTargeting() throws Exception {  
    ConsumerData male = new ConsumerData("000005")  
                                .withGender(Gender.MALE);  
    consumerScenarios.signUp(male);  
    ConsumerData female = new ConsumerData("000006")  
                                .withGender(Gender.FEMALE);  
    consumerScenarios.signUp(female);  
  
    //Add campaign targeting males  
    String maleName = "GenderMale";  
    CampaignData maleCampaign = new CampaignData(mediaAgency,  
                                                    maleName);  
    maleCampaign.targetData = new TargetData().withGender(Gender.MALE);  
    maleCampaign.addDefaultWapBanner();  
    campaignScenarios.addCampaign(maleCampaign);  
}
```


Increasing Readability with a DSL



- Business people can't usually read Java



MultiParameterWapAdTargeting

Edit

Properties

Refactor

Where Used

Search

Files

Versions

Recent Changes

User Guide

Contents:

`com.drutt.test.aop.businessrules.WapAdTargeting`

This test checks some made up targeting parameters

Operator

name	max SMS	max MMS	max Tot
WapOperator	5	5	10

Inventories

name	interests	blacklisted
General	General	-
HairCare	Beauty, Hair	-

Consumers

msisdn	gender	shoe size	dress size	hair length
000001	male	43	-	short
000002	female	40	40	long



Edit

Properties

Refactor

Where Used

Search

Files

Versions

Recent Changes

User Guide

Consumers

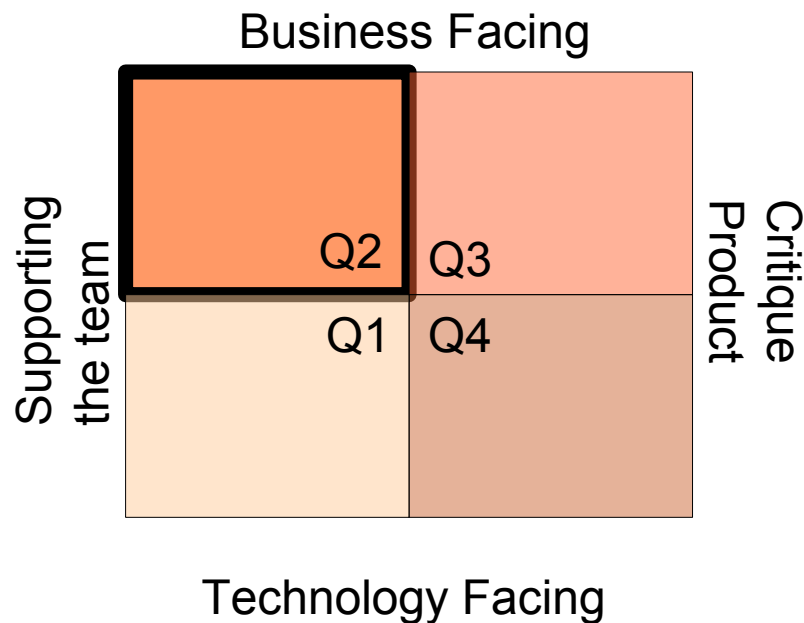
msisdn	gender	shoe size	dress size	hair length
000001	male	43	-	short
000002	female	40	40	long
000003	female	38	-	short
000004	female	40	40	long
000005	female	39	46	short
000006	female	40	42	long

Campaigns

Name	gender	shoe size	dress size	hair length
NoParticularTarget	-	-	-	-
TargetsMale	male	-	-	-
TargetsSmall	female	30-38	34-40	-
HairProduct	female	-	-	long
HairProductShort	-	-	-	short

check	for inventory	General	msisdn	000001	gets WAP ad from campaign	TargetsMale
check	for inventory	General	msisdn	000002	gets WAP ad from campaign	HairProduct
check	for inventory	General	msisdn	000003	gets WAP ad from campaign	TargetsSmall
check	for inventory	General	msisdn	000004	gets WAP ad from campaign	HairProduct

Selenium is only one part of your test strategy



- Tests against a UI are relatively expensive to maintain



Conclusions

- You can test your web application using Selenium
- There are lots of things you can do to increase test:

Readability



Robustness



Speed



Coverage



- Organize test code using domain scenarios or PageObjects



Questions?

- <http://seleniumhq.org/>
- <http://www.xeolabs.com/portal/articles/selenium-and-extjs>
- <http://emilybache.blogspot.com>
- <http://jsolutions.se>

I would like to thank Kamilla Asp from Ericsson for allowing me to use Ad Orchestrator to illustrate my talk with examples.

Images attribution: DaPino Webdesign, Lebreton, Asher Abbasi, Woothemes, Asher, Iconshock, Andy Gongea, from www.iconspedia.com