

White-Box Testing Practice

With Powerful Java Testing Tools

Forest Xie
Nexus International Team

JUnit



Dec 2011



Agenda



**White box
testing basic**



**White testing
in Nexus**

JUnit

**Junit Testing
practice**

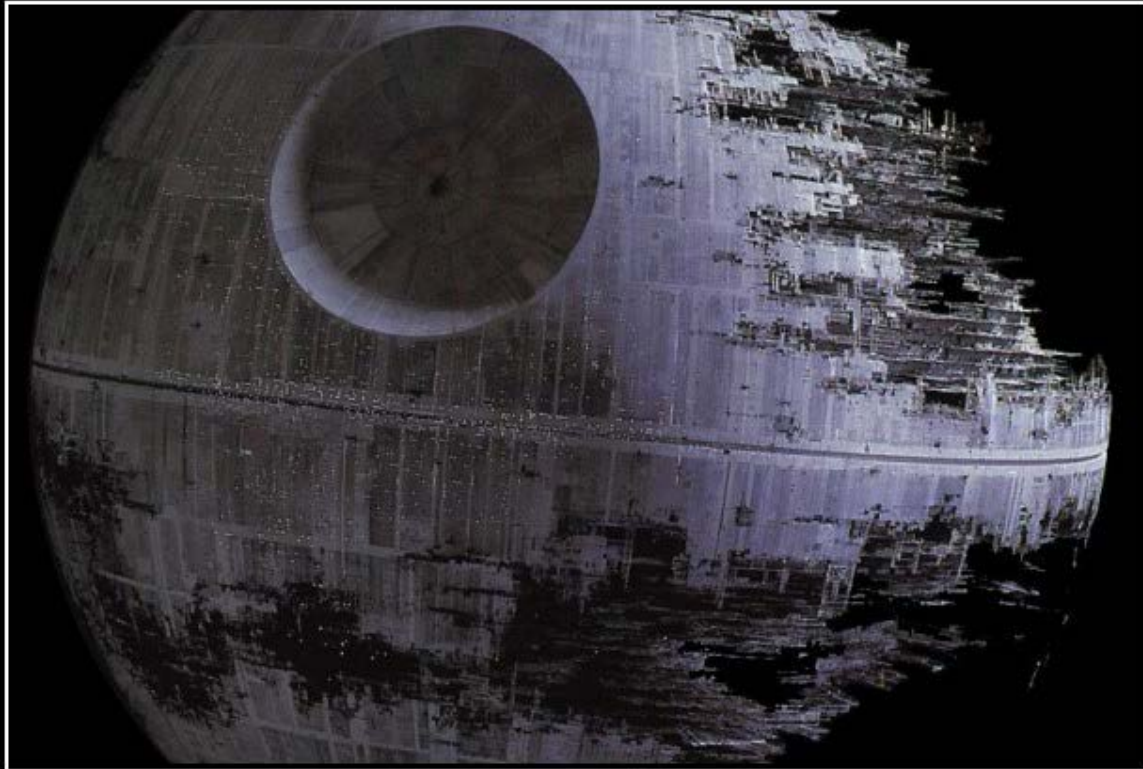


**IT Testing
practice**



**Behave Testing
practice**

Do you have the same feeling?

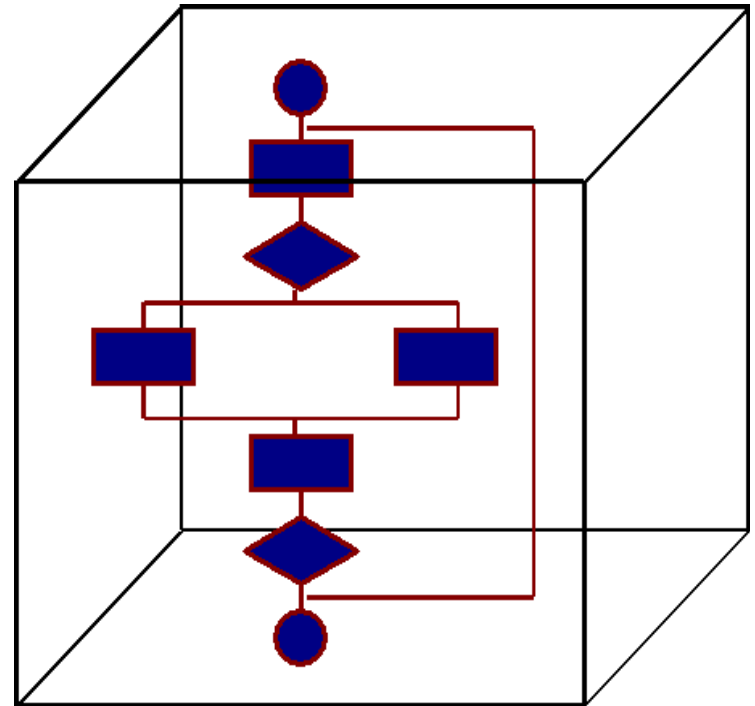


TESTING

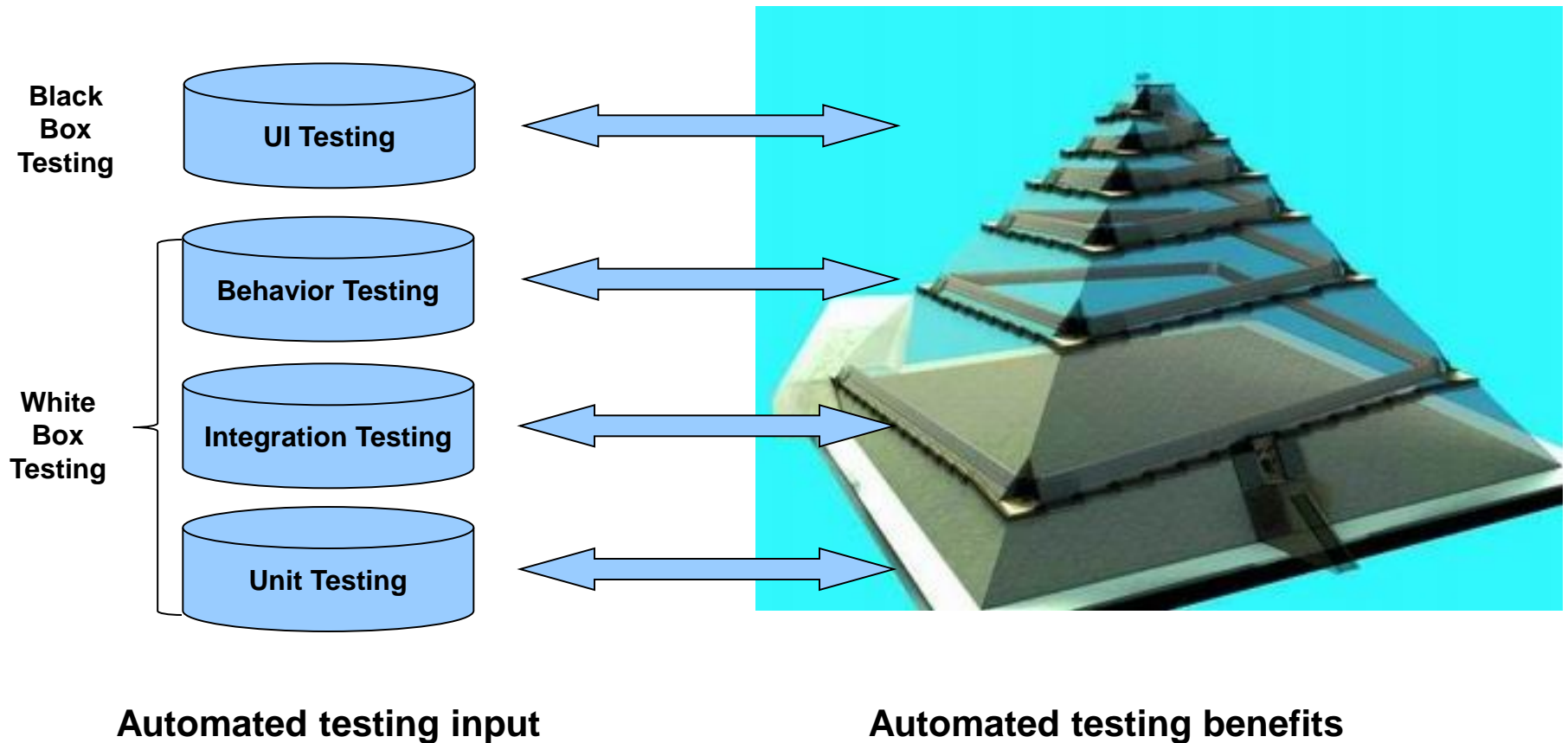
DON'T BE TOO PROUD OF THIS TECHNOLOGICAL TERROR YOU'VE
CONSTRUCTED. THE ABILITY TO DESTROY A PLANET IS INSIGNIFICANT NEXT
TO THE POWER OF TESTING.

About White box testing

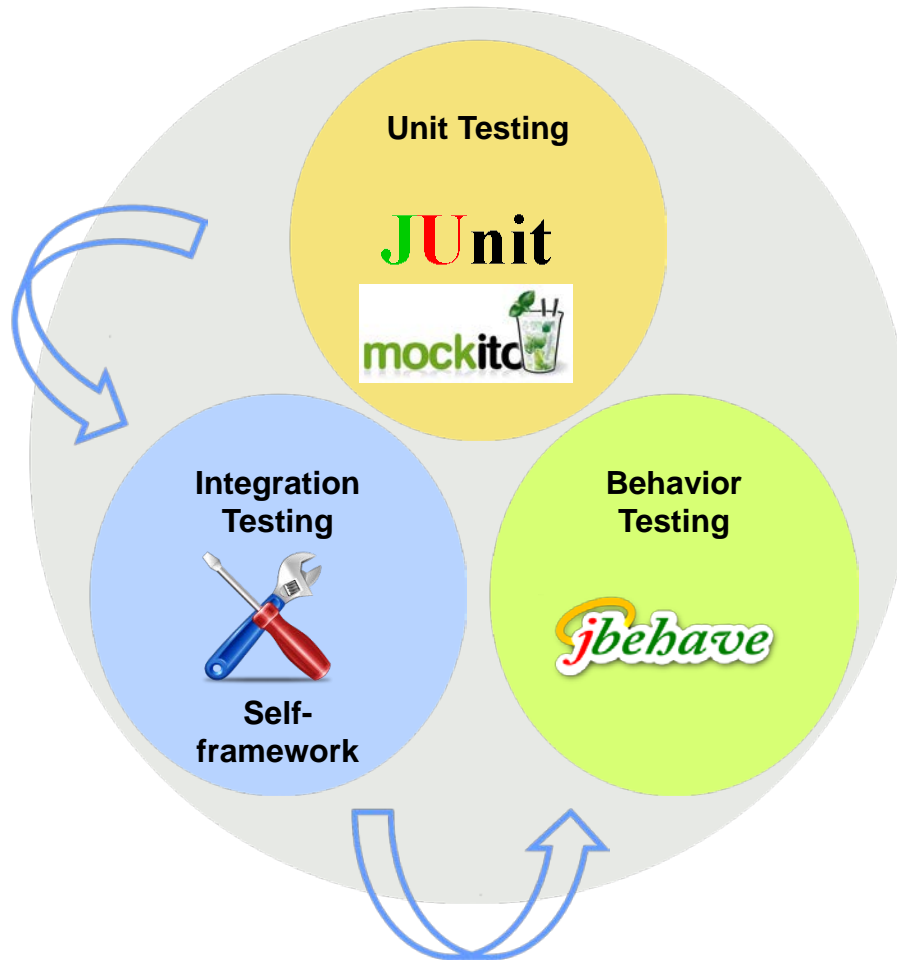
- Testing the internal structure of the software
- Understand the code and the code will set you free!
- White-box testing is testing that takes into account the internal mechanism of a system or component
- It's also known as Structural testing, clear box testing and glass box testing






Why need White box testing



How to implement WBT for us



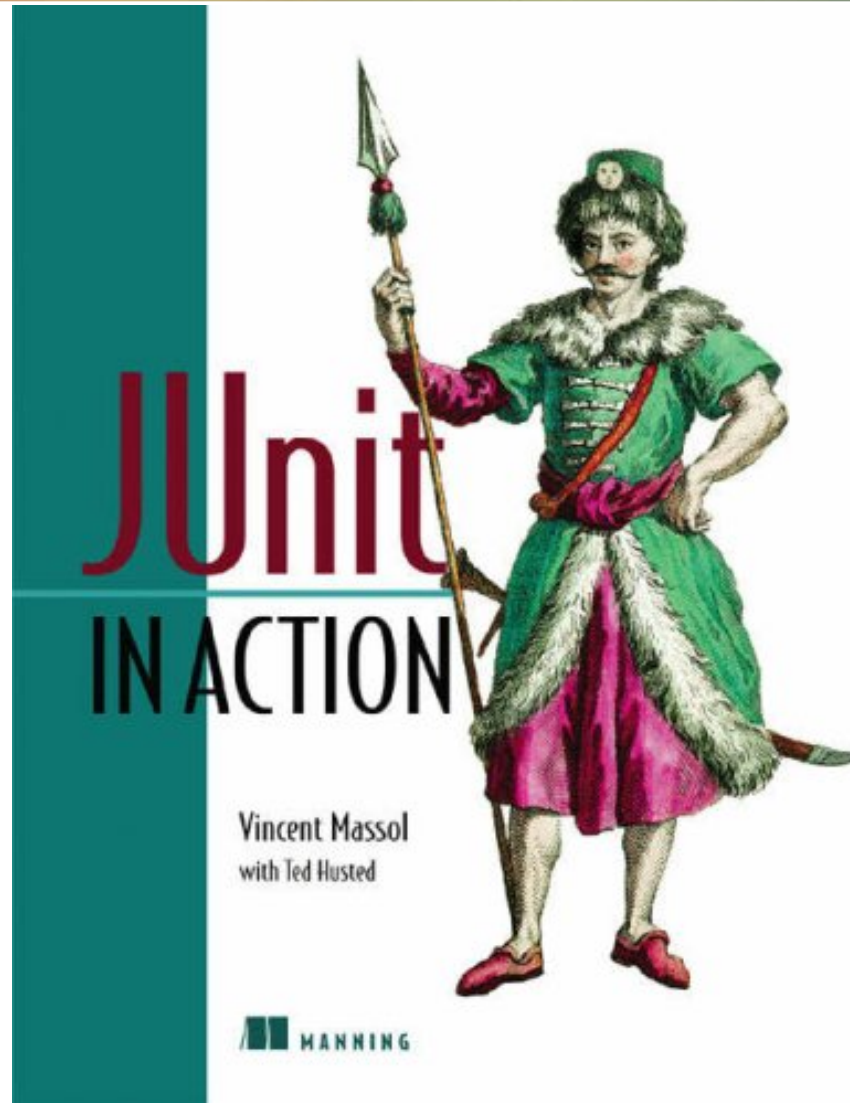
Improving your testing game

- ✓ Innovative testing techniques 
- ✓ Automating unit tests 
- ✓ Separate unit tests and integration tests
- ✓ Monitor test duration 
- ✓ Keep tabs on test coverage

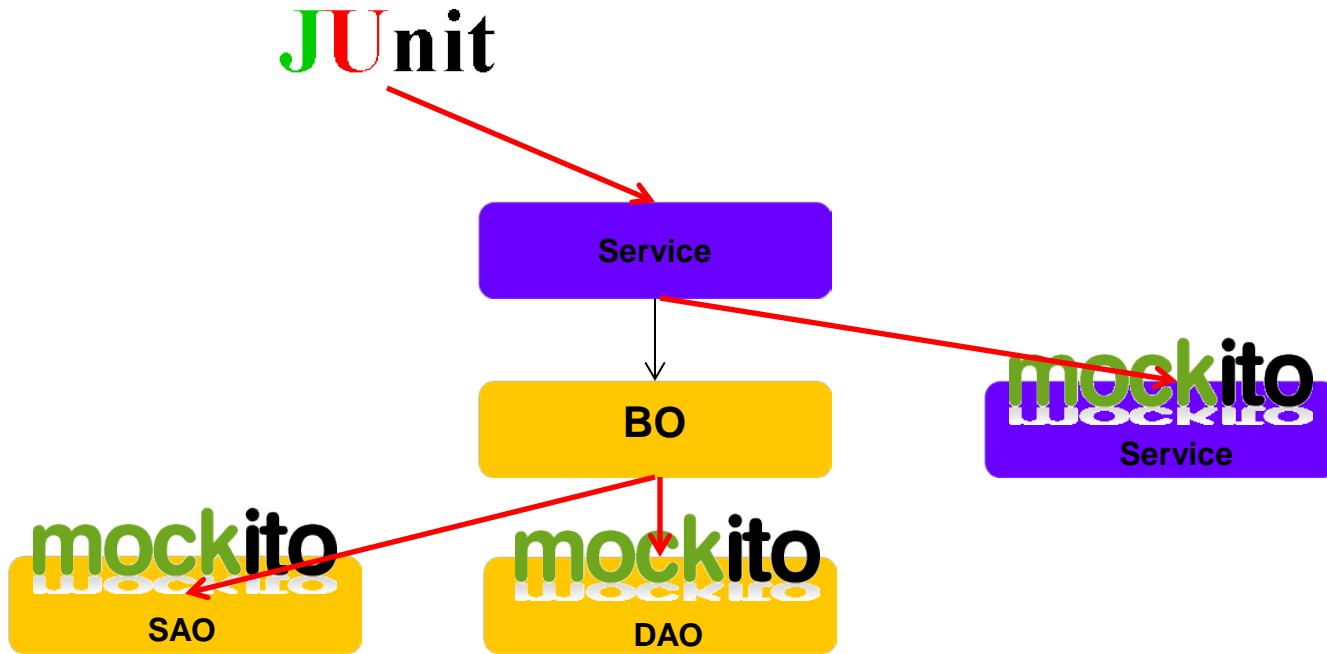


Testing strategy level by level

	Unit-Testing	Integration-Testing	Behavior-Testing
Testing Range	Method, Class	Controller, Business Process	Story from PM
The risk concerned about	The logic error in one Class	Integration issues of multiple modules, some system integration configuration	The feature has matched with PM's requirement
Execution environment	PD's local machine	An integrated test environment	An integrated test environment
Test Type	Out-Container	In-Container	In-Container
Database	N/A	Testing Data	Testing Data



How to develop JUnit testing cases



Use mocks and stubs

Mockito - lightweight mocking



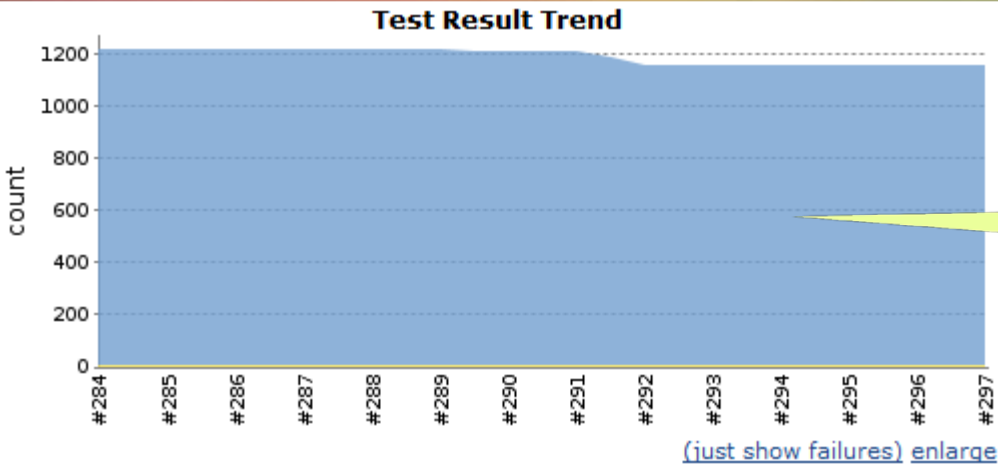
```
@Test
public void stubbing() {
    Stack<String> mockStack = mock(Stack.class);
    when(mockStack.pop()).thenReturn("hi there!");
    assertThat(mockStack.pop(), is("hi there!"));
}
```

Define Stubs

```
@Test
public void verifyCollaboration() {
    Stack<String> mockStack = mock(Stack.class);
    // do stuff
    mockStack.push("hi there!");
    mockStack.pop();
    // check what happened
    verify(mockStack).push("hi there!");
    verify(mockStack).pop();
}
```

**Verify
interactions**

Integrated with Hudson



Unit testing trend by builds

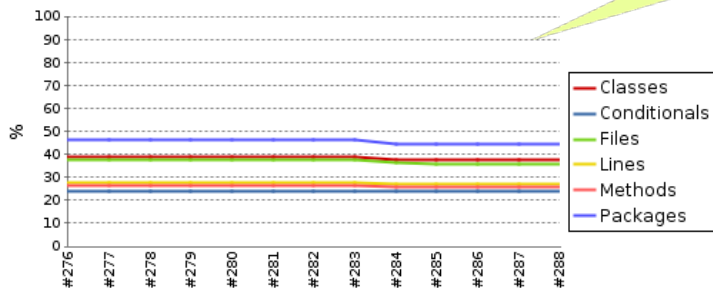
Unit testing Report

1,212 tests (±0)				
Module	Fail	(diff)	Total	(diff)
com.ebay.belen.core:belen-core-ad-mgmt	0		99	
com.ebay.belen.core:belen-core-common	0		122	
com.ebay.belen.core:belen-core-facebook-integration	0		3	
com.ebay.belen.core:belen-core-features	0		10	
com.ebay.belen.core:belen-core-mail	0		6	
com.ebay.belen.core:belen-core-payment	0		40	
com.ebay.belen.core:belen-core-popsearch	0		6	
com.ebay.belen.core:belen-core-search	0		49	
com.ebay.belen.core:belen-core-tns	0		58	
com.ebay.belen.core:belen-core-user	0		38	
com.ebay.belen.core:belen-core-web	0		86	
com.ebay.belen.germany:belen-germany-adverts-adapter-webapp	0		4	
com.ebay.belen.germany:belen-germany-api	0		112	
com.ebay.belen.germany:belen-germany-application	0		94	
com.ebay.belen.germany:belen-germany-batch	0		53	
com.ebay.belen.germany:belen-germany-common	0		34	
com.ebay.belen.germany:belen-germany-content-rules	0		6	
com.ebay.belen.germany:belen-germany-features	0		6	
com.ebay.belen.germany:belen-germany-mail	0		3	
com.ebay.belen.germany:belen-germany-migration	0		16	
com.ebay.belen.germany:belen-germany-solr	0		10	
com.ebay.belen.germany:belen-germany-tns	0		27	
com.ebay.belen.germany:belen-germany-tns-consumer-app	0		3	
com.ebay.belen.germany:belen-germany-tns-webapp	0		14	
com.ebay.aem.hiv.australia:hiv-australia-webapp	0		311	

Uniting testing code coverage report

Cobertura Coverage Report

Trend



The trend for unit testing code coverage

The detail for unit testing code coverage

Project Coverage summary

Name	Classes	Conditionals	Files	Lines	Methods	Packages
Cobertura Coverage Report	38% 817/2162	24% 4867/20572	36% 624/1734	27% 14960/54971	26% 3786/14776	44% 216/486

Coverage Breakdown by Package

Name	Classes	Conditionals	Files	Lines	Methods
com.ebay.belen.germany.tns.replyblock.model	0% 0/1	N/A	0% 0/1	0% 0/8	0% 0/4
com.ebay.belen.webapp.adapter.yelp.api	0% 0/1	N/A	0% 0/1	0% 0/1	0% 0/1
com.ebay.belen.core.popsearch.service.impl	50% 2/4	6% 4/62	50% 2/4	15% 17/114	32% 6/19
com.ebay.belen.core.common.support.reporting	0% 0/2	0% 0/20	0% 0/2	0% 0/75	0% 0/24
com.ebay.belen.germany.webapp.postad.model	83% 5/6	34% 59/174	83% 5/6	33% 97/292	27% 32/119
com.ebay.belen.germany.webapp.postad.controller	75% 12/16	45% 172/380	75% 12/16	58% 364/629	50% 52/104
com.ebay.belen.germany.webapp.advertiser.impl	67% 2/3	N/A	67% 2/3	68% 15/22	62% 8/13
com.ebay.belen.core.tns.flag.dao	0% 0/3	0% 0/7	0% 0/2	0% 0/43	0% 0/16
com.ebay.belen.germany.tns.replyblock.filter.service	0% 0/2	0% 0/16	0% 0/2	0% 0/45	0% 0/11
com.ebay.belen.germany.tns.webapp.webapi.expressions	0% 0/3	0% 0/12	0% 0/3	0% 0/72	0% 0/24

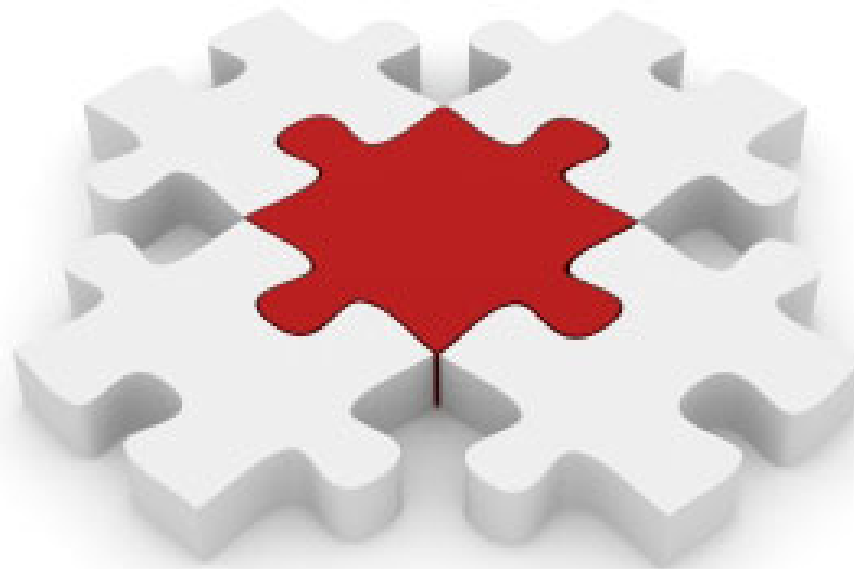
How to get unit testing report in Hudson

- Demo

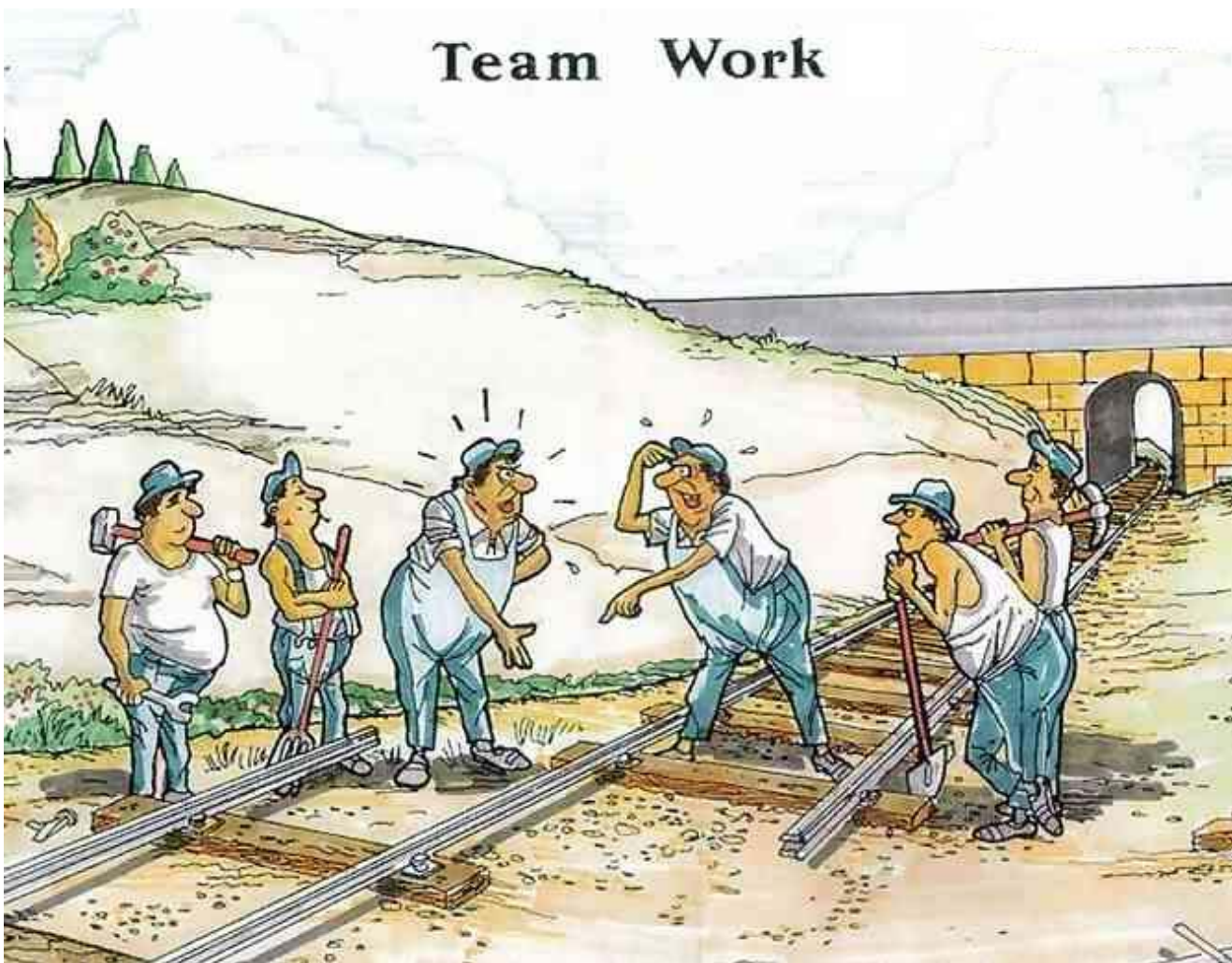


TESTING DO NOT DISTURB

Integration test



If we don't have integration testing...



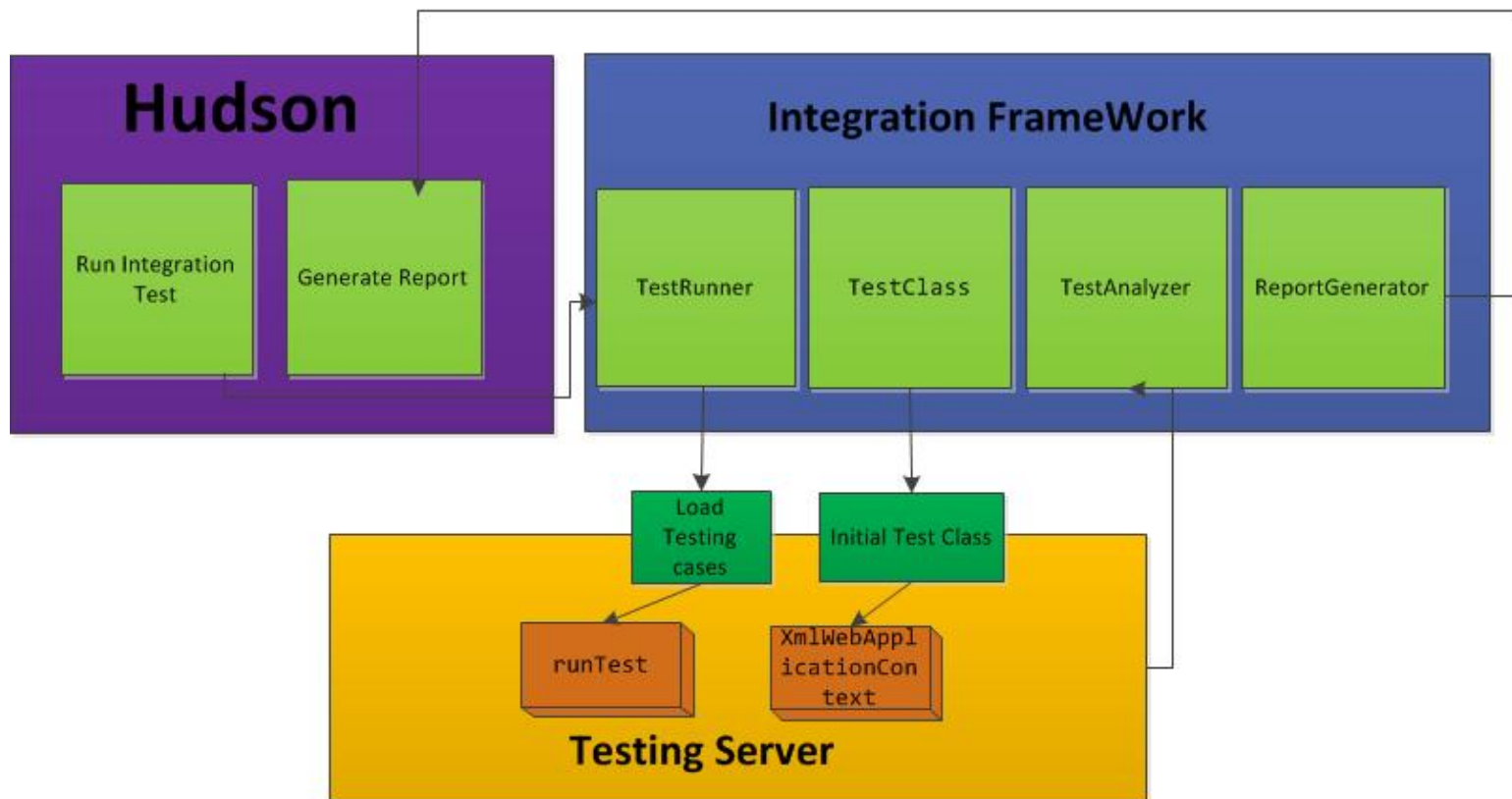
Good Integration framework should be

- ✓hide unnecessary details
- ✓use reusable components
- ✓be low maintenance
- ✓Have a good report
- ✓Can be read by each person



The integration Framework structure

Integration structure



Test cases management

- Test Suite: Each controller Test class has its test suite, all the test cases for the controller defined in the test suite file.

```
<testSuite_name="loginTestSuite" _description="login_integration_testing" _javaClass="com.ebay.kijiji.integration.test.LoginControllerTest" _>$
$
```

- Test Service: Each method of controller has its own test service. It can include multiple test cases.

```
<service_name="loginService" _methodName="loginTest" _inputSet="loginSet">$
  <rule_name="validRule">$
    <field_name="validErrorSize">$
      <condition_name="equals" _/>$
    </field>$
    <field_name="Field">$
      <condition_name="equals" _/>$
    </field>$
    <field_name="ErrorCode">$
      <condition_name="equals" _/>$
    </field>$
    <field_name="RejectValue">$
      <condition_name="equals" _/>$
    </field>$
  </rule>$
```

- Test cases: all the input setting are defined in a test case.

```
<testCase_name="loginCaseWithRightAccountAndPassword" _description="login_with_right_account_and_password">$
  <param name="email" _value="test@mail.de" _type="java.lang.String" _/>$
  <param name="password" _value="3453$$wfsdfWER" _type="java.lang.String" _/>$
</testCase>$
```

Add integration test cases

```
<?xml_version="1.0" _encoding="GB2312"?>$
<testSuite_name="loginTestSuite" _description="login_integration_testing" _javaClass="com.ebay.kijiji.integration.test.LoginControllerTest">$
<!-- ////////////////////////////////////// _-->$
<!-- TEST_INPUTS _-->$
<!-- ////////////////////////////////////// _-->$
<input>$
  <inputSet_name="loginSet">$
    <testCase_name="loginCaseWithRightAccountAndPassword" _description="login_with_right_account_and_password">$
      <param name="email" _value="test@mail.de" _type="java.lang.String" _/>$
      <param name="password" _value="3453$$s$wfsdfWER" _type="java.lang.String" _/>$
    </testCase>$
    <testCase_name="loginCaseWithUnregisteredAccount" _description="login_account_is_n">$
      <param name="email" _value="doxie@ebay.com" _type="java.lang.String" _/>$
      <param name="password" _value="123" _type="java.lang.String" _/>$
    </testCase>$
    <testCase_name="loginCaseWithInvalidAccount" _description="login account_email_for">$
      <param name="email" _value="doxie#ebay.com" _type="java.lang.String" _/>$
      <param name="password" _value="123" _type="java.lang.String" _/>$
    </testCase>$
    <testCase_name="loginCaseWithEmptyPassword" _description="password_is_empty">$
      <param name="email" _value="doxie@ebay.com" _type="java.lang.String" _/>$
      <param name="password" _value="" _type="java.lang.String" _/>$
    </testCase>$
    <testCase_name="loginCaseWithSpacePassword" _description="password_is_space">$
      <param name="email" _value="doxie@ebay.com" _type="java.lang.String" _/>$
      <param name="password" _value=" " _type="java.lang.String" _/>$
    </testCase>$
  </inputSet>$
</input>$
</testSuite>$
```

Specified the testing class

Add the test case

Set the inputting parameters

One class, multiple case
All the test cases defined in XML file. Test codes and cases are separated.

Add a executing test class

```
*  
* Copyright (c) Sep 14, 2011  
*  
* @author doxie@ebay.com  
* @version 1.0  
*  
*/  
@Ignore  
public class LoginControllerTest extends TestClass{  
  
    private LoginController loginController = new LoginController();  
    private LoginFormValidator loginFormValidator;  
    private LoginService loginService;  
    private CaptchaManager captchaManager;  
    private BotBlockerService botBlockerService;  
    private IpAddress ipAddress = new IpAddress("127.0.0.1");  
    private String targetUrl = "/m-meine-anzeigen.html";  
  
    @Override  
    public void setUp() {  
        this.loginFormValidator = WebappControllerTest.getBeanByType(LoginFormValidator.class);  
        this.captchaManager = WebappControllerTest.getBeanByType(CaptchaManager.class);  
        this.loginService = WebappControllerTest.getBeanByType(LoginService.class);  
        this.botBlockerService = WebappControllerTest.getBeanByType(BotBlockerService.class);  
        ReflectionTestUtils.setField(loginController, "loginFormValidator", loginFormValidator);  
        ReflectionTestUtils.setField(loginController, "loginService", loginService);  
        ReflectionTestUtils.setField(loginController, "captchaManager", captchaManager);  
        ReflectionTestUtils.setField(loginController, "botBlockerService", botBlockerService);  
    }  
}
```

Inherited
TestClass

Initialized the service and
manager

Add testing method in test class

```
public void loginTest(String email, String password) throws Exception
{
    setUp();
    LoginForm loginForm = loginController.createLoginForm(ipAddress, email, targetUrl);
    loginForm.setPassword(password);
    BindingResult bindingResult = new BeanPropertyBindingResult(loginForm, "loginForm");
    HttpServletRequest request = new MockHttpServletRequest();
    HttpServletResponse response = new MockHttpServletResponse();
    BelenRequestWrapper wrapper = new BelenRequestWrapper(request);
    ModelMap map = new ModelMap();
    String urlString = loginController.login(loginForm, bindingResult, map, ipAddress, wrapper, response);

    TestResult result = getTestLogger().newTestResultInstance();

    TestResultRecord validRecord = new TestResultRecord("validRule");
    int count = bindingResult.getErrorCount();
    validRecord.addResult(new TestValue("validErrorSize", count));
    if(count > 0 )
    {
        for(FieldError error : bindingResult.getFieldErrors())
        {
            validRecord.addResult(new TestValue("Field", error.getField()));
            validRecord.addResult(new TestValue("ErrorCode", error.getCode()));
            validRecord.addResult(new TestValue("RejectValue", error.getRejectedValue().toString()));
        }
    }
    result.addRecord(validRecord);
    TestResultRecord commanRecord = new TestResultRecord("loginControllerRule");
    boolean authError = (Boolean)(map.get("authError") != null?map.get("authError"):false);
    boolean LoginSuccessfully = (authError == false && count == 0);
    commanRecord.addResult(new TestValue("urlString", urlString));
    commanRecord.addResult(new TestValue("LoginSuccessfully", LoginSuccessfully));
    result.addRecord(commanRecord);
    getTestLogger().saveTestResult(result);
}
```

Execute Controller

Collect result into
TestResult

Integration report

Testing Suite

No.	Test Suites	Status
1.	homePageControllerSuite	pass
2.	loginTestSuite	pass
3.	mytestsuite	pass

All Test Suites -> [loginTestSuite](#) -> loginService

Test Service	loginService
Method Name	loginTest
Description	

No.	Test Cases	Description	Status
1.	loginCaseWithEmptyPassword	password is empty	pass
2.	loginCaseWithInvalidAccount	login account email format is not valid	pass
3.	loginCaseWithRightAccountAndPassword	login with right account and password	pass
4.	loginCaseWithSpacePassword	password is space	pass
5.	loginCaseWithUnregisteredAccount	login account is not exist	pass

Testing Services

Integration report

All Test Suites -> [searchTestSuite](#) -> [searchService](#) -> searchWithCategory

Test Case	searchWithCategory
Description	search with automotive
Reference Date	11-11-1 下午3:26
Test Date	11-11-1 下午4:35

Testing Input

Input		
Field Name	Value	Type
keyWords	car	java.lang.String
categoryId	9299	long
locationStr	Inner Sydney	java.lang.String
distance	5	long

Output				
Record 1				
Field Name	Ref Value	Test Value	Condition(s)	Status
name	Kategorien	Kategorien	==	pass
Record 2				
Field Name	Ref Value	Test Value	Condition(s)	Status
name	Alle Kategorien	Alle Kategorien	==	pass
Count	33708	1432	==	fail
SeoUrl	/s-car/k0r5	/s-inner-sydney/car/k0l3003771r5	==	fail

Testing result

Integrated with Hudson

Hudson » [box-trunk-QA-integration-testing](#)

-  [返回](#)
-  [状态](#)
-  [变更集](#)
-  [工作区](#)
-  [立即构建](#)
-  [删除Project](#)
-  [设置](#)
-  [Delete All Disabled Modules](#)
-  [Modules](#)
-  [Integration-Testing Report](#)

Project box-trunk-QA-integration-testing

execute all the Integration testing and verify if all the cases are pass



[Integration-Testing Report](#)



[Workspace](#)



[Recent Changes](#)

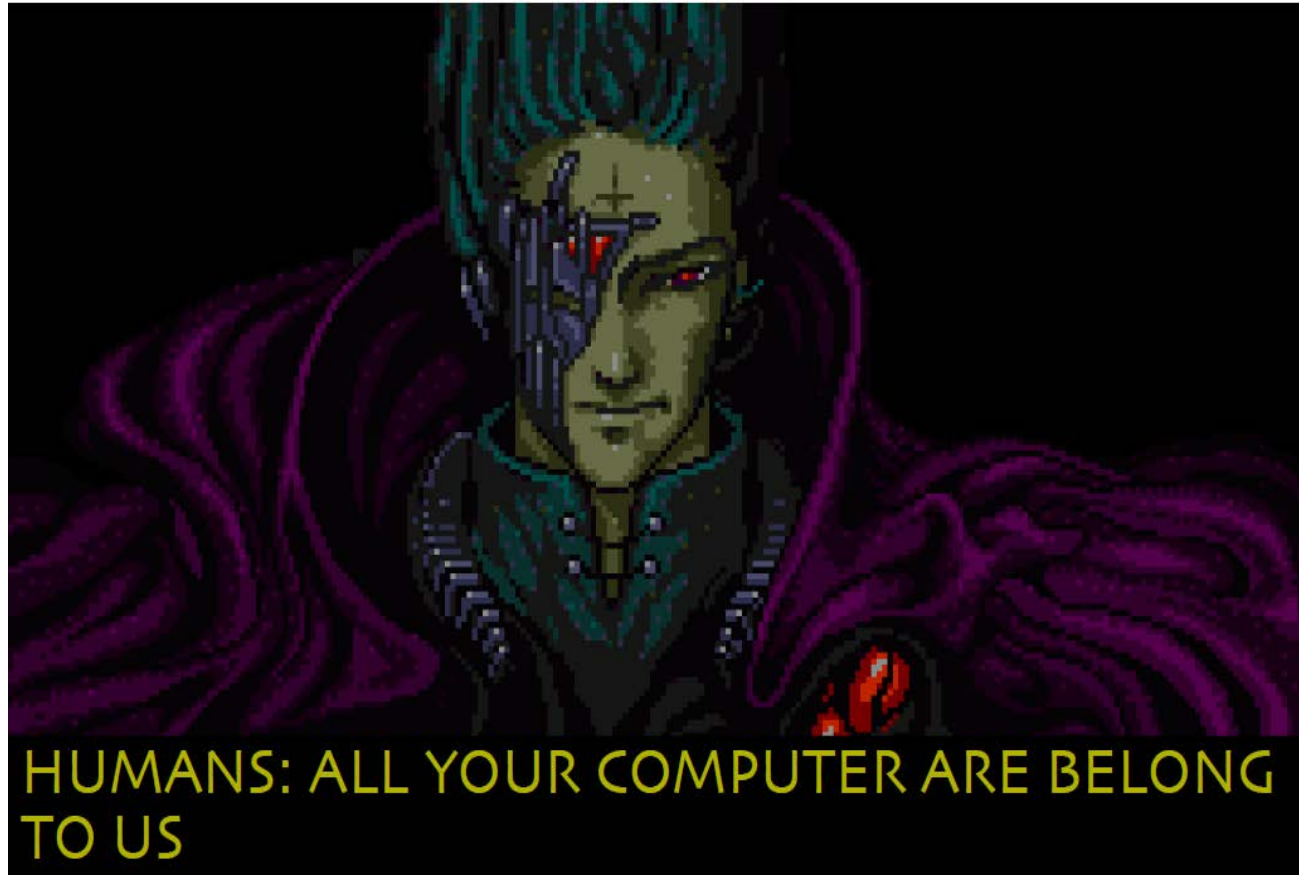
We can executed all the integration test by hudson

[Back to box-trunk-QA-integration-testing](#) [index](#)

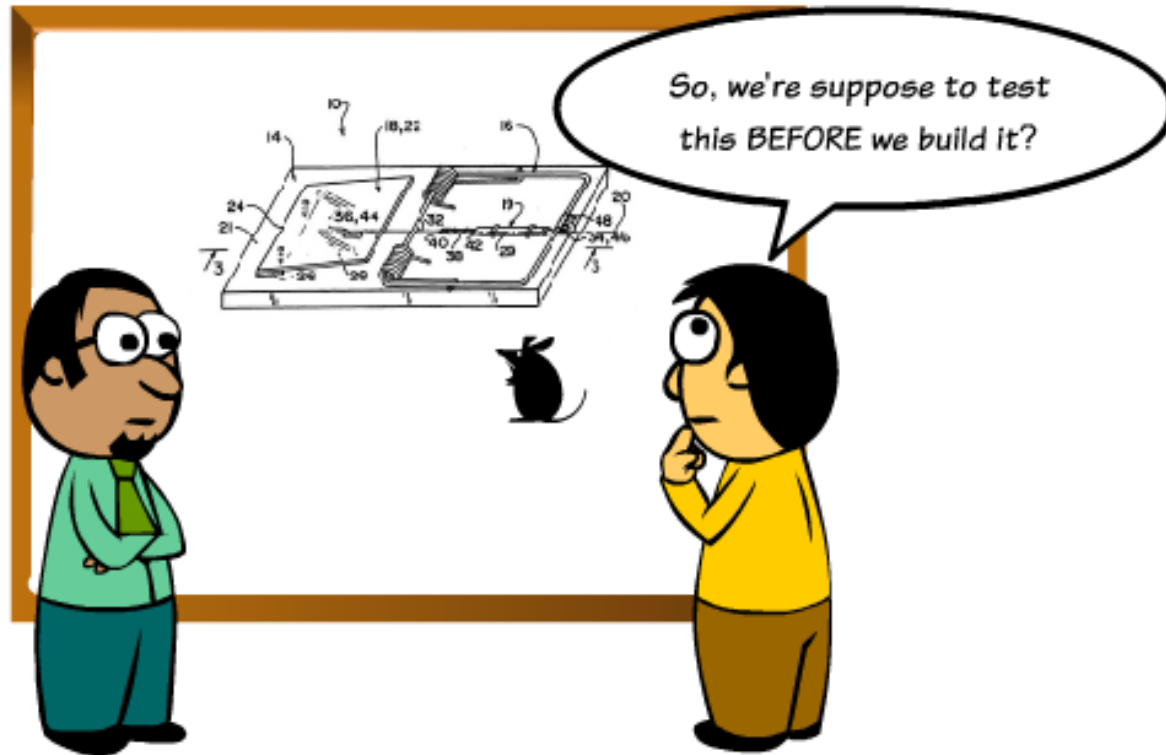
[Zip](#)

No.	Test Suites	Status
1.	homePageControllerSuite	pass
2.	loginTestSuite	pass
3.	mytestsuite	pass

- Demo

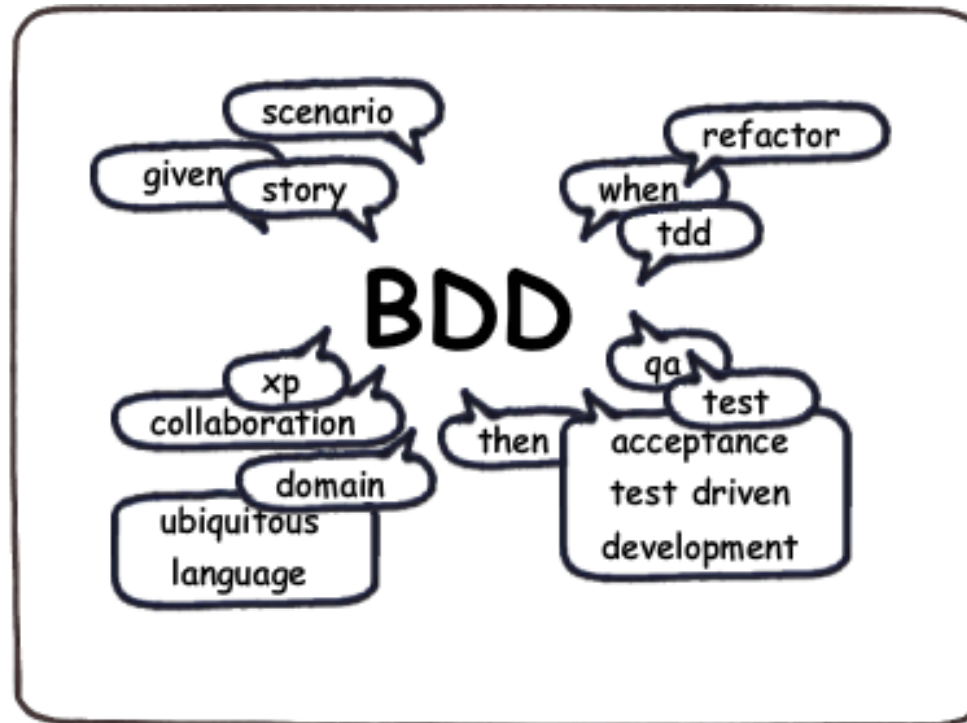


Behavior Test



Behavior Test Driven Development

- BTDD (Behavior Test Driven Development) is the extension of TDD, which helps us deliver exactly what the PM/ Customer wants.
- Its strongest point is that the business analyst is the root of the process.



how to use the JBehave in our project

Spec/Scenario	• Natural language
Parser	• JBehave
Step Logic	• TestSteps[POJO]
Action/utils	• Action/Utils(Reusable)

Specification/Scenario layer:

This layer describes system's behaviors and functionalities by the scenarios. For using JBehave, we can use the natural language describe the scenarios and just need to follow the JBehave 'Given-When-Then' rule.

Parser layer:

We don't need to implement this layer , this layer has been implemented by JBehave. What exactly JBehave do is to relate the steps of the scenario to the methods of the test codes.

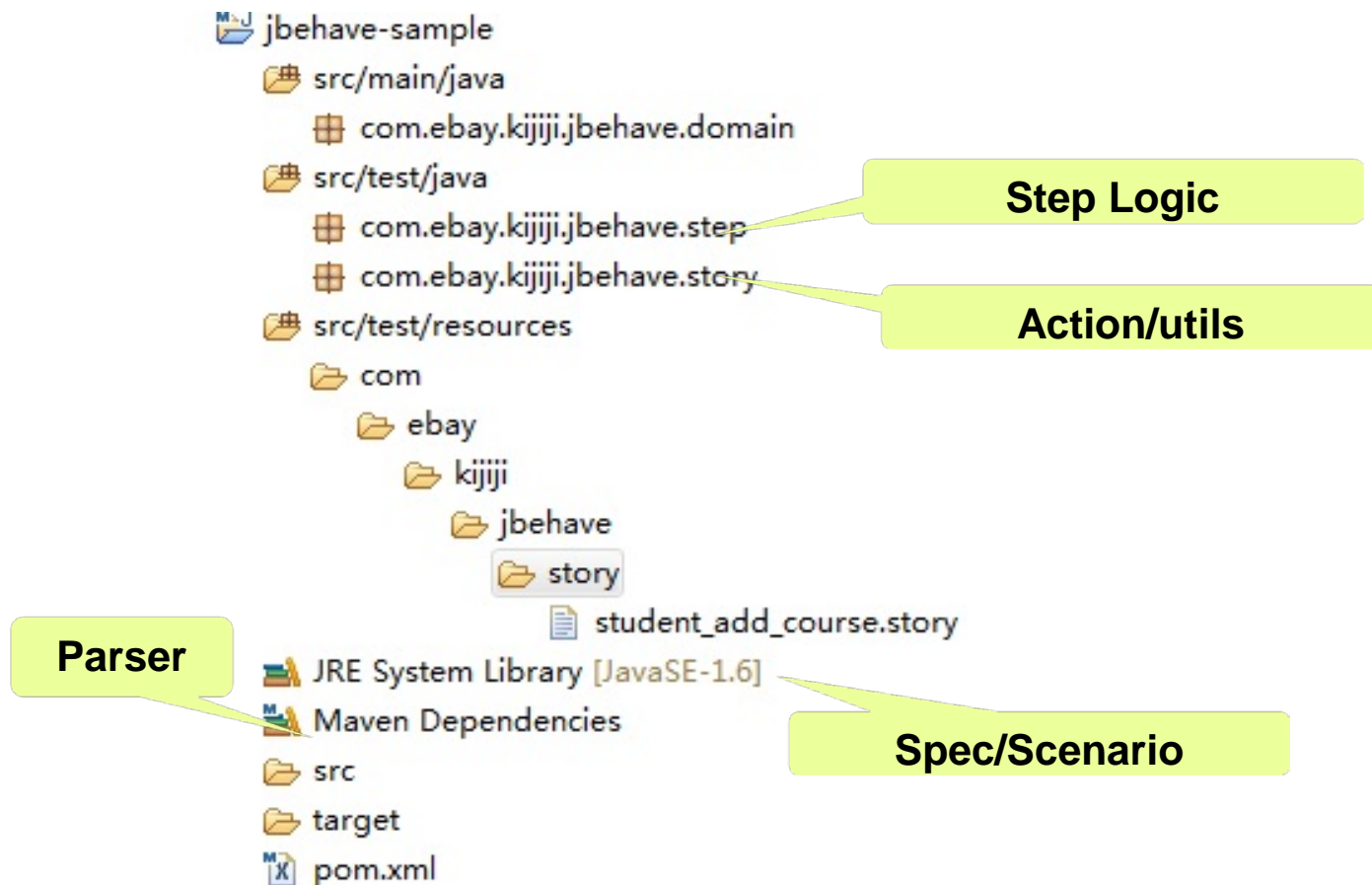
Step Logic Layer:

The layer implements test logics associating with every step of the scenarios. Every step are implemented by a Java method.

Action/Utils layer

This the very important layer to improve the reusability of our codes. This layer provides the utility methods to help you implement step logics. These utility methods usually involved the system state checking, mock requests sending and so on.

The project structure in eclipse



Jbehave Report

Story Reports

Test Class

Name	Statistics	View
com.ebay.kijiji.jbehave.story.student_add_course	Scenarios: 1 (Failed: 0) Steps: 4 (Successful: 4; Pending: 0; Not Performed: 0; Failed: 0; Ignorable: 0;)	stats txt

Generated at 03/10/2011 17:58:01
JBehave 3.x © 2003-2010

Test result

com/ebay/kijiji/jbehave/story/student_add_course.story

Scenario: when student registered to a course, course has said student

Given a student
 Given a course
 When the student registers in the course
 Then the course must have the student as one of its student

Jbehave Code Coverage report

file:///D:/SHC-00426404/Users/doxie/Downloads/jbehave-example/target/site/cobertura/index.html

火狐官方网站 访问最多 新手上路 最新头条 网址大全

Packages
[All](#)
[com.ebay.kijiji.jbehave.domain](#)

Coverage Report - com.ebay.kijiji.jbehave.domain

Package	# Classes	Line Coverage	Branch Coverage	Complexity
com.ebay.kijiji.jbehave.domain	4	45% 17/37	N/A	N/A

Classes in this Package	Line Coverage	Branch Coverage	Complexity
Course	50% 7/14	N/A	N/A
Person	66% 4/6	N/A	N/A
Student	66% 6/9	N/A	N/A
Teacher	0% 0/8	N/A	N/A

Report generated by [Cobertura](#) 1.9.4.1 on 11-10-4 下午1:10.

com.ebay.kijiji.jbehave.domain

Classes
[Course](#) (50%)
[Person](#) (66%)
[Student](#) (66%)
[Teacher](#) (0%)

file:///D:/SHC-00426404/corp.ebay.com/share/site/cobertura/index.html

火狐官方网站 访问最多 新手上路 最新头条 网址大全

Packages
[All](#)
[com.ebay.kijiji.jbehave.domain](#)

Coverage Report - com.ebay.kijiji.jbehave.domain.Student

Classes in this File	Line Coverage	Branch Coverage	Complexity
Student	66% 6/9	N/A	N/A


```

1 package com.ebay.kijiji.jbehave.domain;
2
3 import java.util.Collections;
4 import java.util.HashSet;
5 import java.util.Set;
6
7 public class Student extends Person {
8
9     private Set<Course> courses = new HashSet<Course>();
10
11     public Student(String firstName, String lastName) {
12         super(firstName, lastName);
13     }
14
15     public void addCourse(Course course) {
16         courses.add(course);
17     }
18
19     public void addStudent(this);
20
21     public Set<Course> getCourses() {
22         return Collections.unmodifiableSet(courses);
23     }
24
25     public void removeCourse(Course course) {
26         courses.remove(course);
27     }
28
29 }
          
```

Report generated by [Cobertura](#) 1.9.4.1 on 11-10-4 下午1:10.

com.ebay.kijiji.jbehave.domain

Classes
[Course](#) (50%)
[Person](#) (66%)
[Student](#) (66%)
[Teacher](#) (0%)

coverage detail
information

Jbehave integrated with Hudson

▼ Git
● Subversion
Modules

Repository URL

Local module directory (optional)

Repository depth option

Ignore externals option ☐

Check-out Strategy

Repository browser

Build Triggers

☐ Build after other projects are built

☐ Poll SCM

☐ Build periodically

☐ Build when Maven dependencies have been updated by Maven 3 integration

☐ Build when Maven SNAPSHOT dependencies have been updated externally

Build

Execute shell

Command

Added Svn address

Executed by maven

• Demo



I FIND YOUR LACK OF
TESTING DISTURBING

Forest Xie

Email: doxie@ebay.com

Skype: [dongsheng.xie](https://www.skype.com/people/dongsheng.xie)

