

Tester une application

HTML



Live coding sur les tests d'acceptances d'une application html5/javascript [#bdd](#) [#javascript](#)

Jean-Laurent de Morlhon

directeur technique

Xebia
Studio



@morlhon

jeanlaurent@morlhon.net



Jeff Gilfelt

@readyState



Developer job title cheat sheet:

Senior = Old

Manager = Expendable

Architect = Can't code

Evangelist = Can't ship

2:59 PM - 25 Apr 2013

281 RETWEETS **66** FAVORITES





DEVELOPMENT DRIVEN TESTS



TESTING

I FIND YOUR LACK OF TESTS DISTURBING.

Copyrighted Material

The Addison-Wesley Signature Series



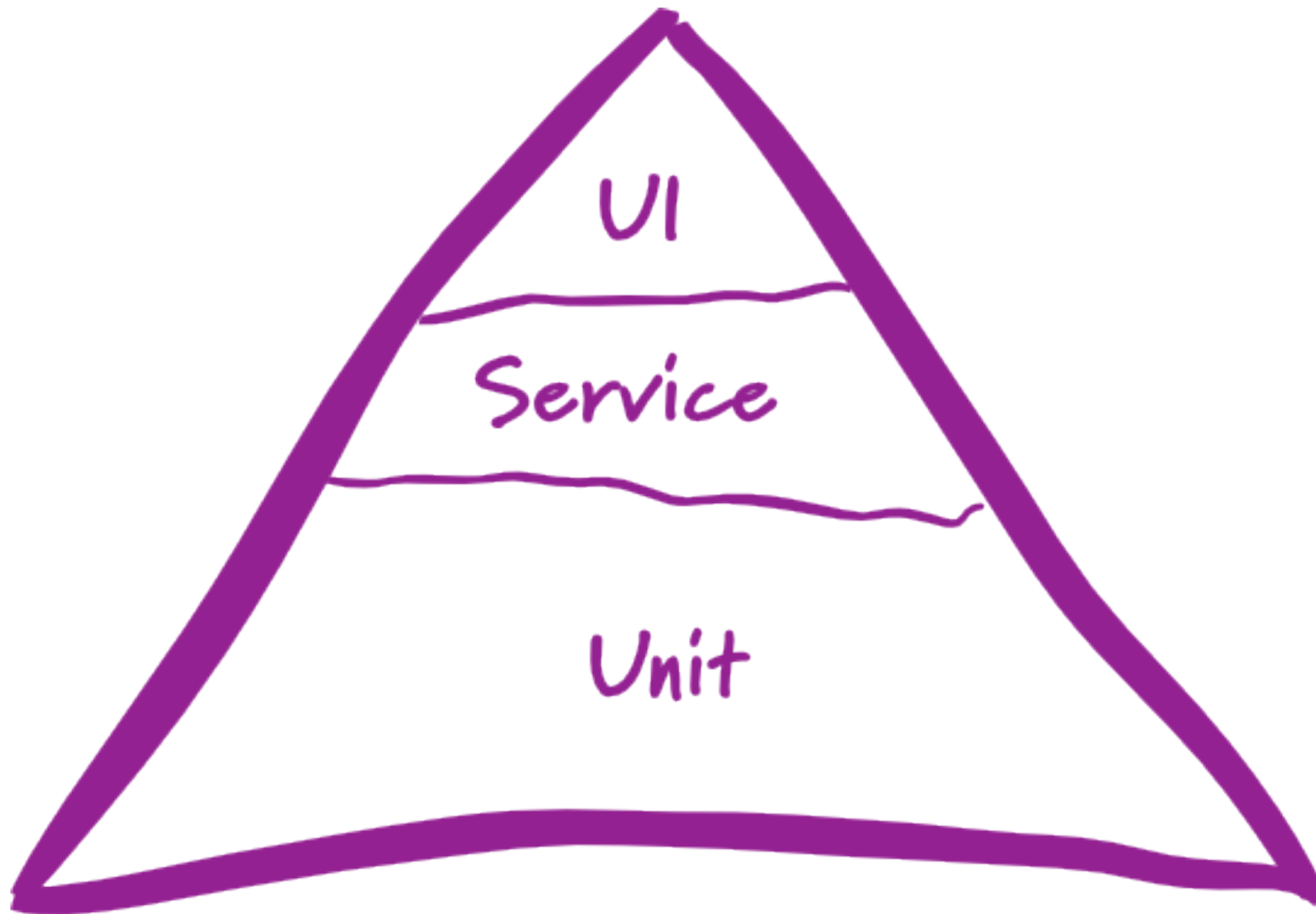
GROWING OBJECT-ORIENTED SOFTWARE, GUIDED BY TESTS

STEVE FREEMAN
NAT PRYCE



Copyrighted Material

Copyrighted Material



Abbot

```
public class LabeledListTest extends ComponentTestFixture {

    public void testLabelChangedOnSelectionChange() throws Throwable {
        String[] contents = { "one", "two", "three" };
        final LabeledList labeledList = new LabeledList(contents);
        showFrame(labeledList);

        Component list = getFinder().find(new ClassMatcher(JList.class));
        JListTester tester = new JListTester();

        JLabel label = (JLabel) getFinder().find(labeledList, new Matcher() {
            public boolean matches(Component c) {
                return c.getClass().equals(JLabel.class)
                    && c.getParent() == labeledList;
            }
        });

        tester.actionSelectRow(list, new JListLocation(1));

        assertEquals("Wrong label after selection",
            "Selected: two", label.getText());

        tester.actionSelectRow(list, new JListLocation(2));
        assertEquals("Wrong label after selection",
            "Selected: three", label.getText());

        tester.actionSelectRow(list, new JListLocation(0));
        assertEquals("Wrong label after selection",
            "Selected: one", label.getText());
    }

    public LabeledListTest(String name) { super(name); }
}
```




HtmlUnit

```
@Test
public void homePage() throws Exception {
    WebClient webClient = new WebClient();
    HtmlPage page = webClient.getPage("http://www.mix-it.fr");
    assertEquals("Accueil - MIX-IT 2013", page.getTitleText());

    String pageAsXml = page.asXml();
    assertTrue(pageAsXml.contains("<body itemscope"));

    String pageAsText = page.asText();
    assertTrue(pageAsText.contains("Des idées pour tout de suite!"));

    webClient.closeAllWindows();
}
```

Selenium

```
public class NewTest extends SeleneseTestCase {  
    // We create our Selenium test case  
  
    public void setUp() throws Exception {  
        setUp("http://www.mix-it.fr/", "*firefox");  
    }  
  
    public void testNew() throws Exception {  
        selenium.open("/");  
        selenium.type("q", "rocks");  
        selenium.click("btnG");  
  
        selenium.waitForPageToLoad("30000");  
  
        assertTrue(selenium.isTextPresent("ROCKS !"));  
    }  
}
```

Zombie.js

```
var Browser = require("zombie");

browser = new Browser();
browser.visit("http://localhost:3000/", function () {

    browser.fill("email", "jeanlaurent@serpodile.com");
    browser.fill("password", "youDontWantToKnow");

    browser.pressButton("Login", function() {
        assert.ok(browser.success);
        assert.equal(browser.text("title"), "Welcome to Serpodile");
    });

});
```

Zombie.js

```
Browser = require("zombie")

browser = new Browser()
browser.visit "http://localhost:3000/", ->

  browser.fill "email", "jeanlaurent@serpodile.com"
  browser.fill "password", "youDontWantToKnow"

  browser.pressButton "Login", ->
    assert.ok browser.success
    assert.equal browser.text "title" , "Welcome to Serpodile"
```

Cucumber

Feature: Addition

In order to avoid silly mistakes

As a math idiot

I want to be told the sum of two numbers

Scenario Outline: Add two numbers

Given I have entered 5 into the calculator

And I have entered 7 into the calculator

When I press the button

Then the result should be 12 on the screen

Talk

- Action

= S**t

Références

- BDD
 - <http://referentiel.institut-agile.fr/bdd.html>
- Zombie.js
 - <http://zombie.labnotes.org/>
- Cucumber.js
 - <https://github.com/cucumber/cucumber-js>
- Chai.js
 - <http://chaijs.com/>

Source & Slides

<https://github.com/jeanlaurent/CucumberAndZombie>

Pour aller plus loin

- UI Test
 - Karma: <http://karma-runner.github.io>
 - Casper.js: <http://casperjs.org/>
 - FluentLenium: <https://github.com/FluentLenium/>
- Unit Testing
 - Jasmine: <https://github.com/pivotal/jasmine>
 - Buster.js : <http://docs.busterjs.org/en/latest/>
 - Mocha : <https://github.com/visionmedia/mocha>
 - QUnit : <http://qunitjs.com/>
 - Sinon.js: <http://sinonjs.org/>

