Test Driven Development for Web Applications

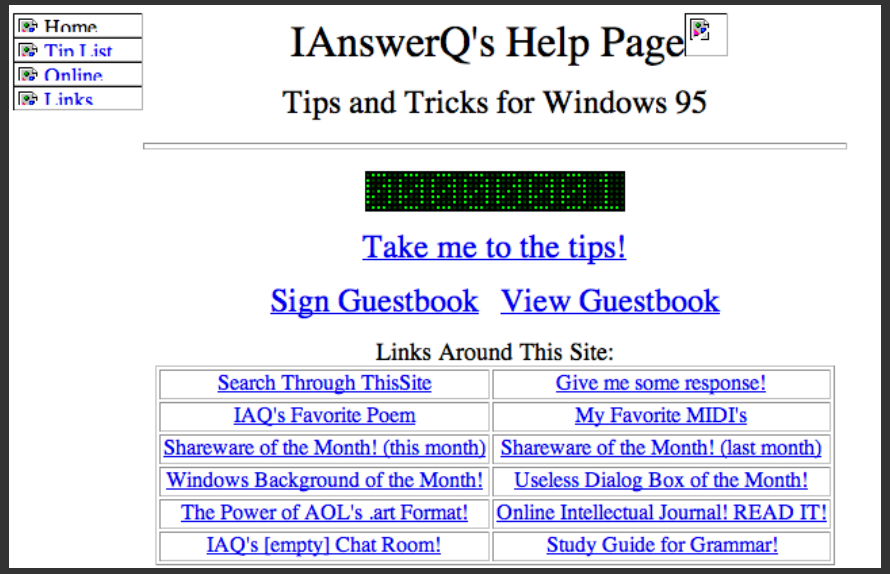
Brett Bieber
brett.bieber@gmail.com
University of Nebraska-Lincoln

2008-10-07 HighEdWeb 2008

About Me

University of Nebraska-Lincoln
University Communications
Web Programmer

My first web page, circa 1995



ASP from 2001-2003
PHP since 2001

- PHP PEAR Project
- W3C HTML Validator
- PHPSavant Templating System

Topics

- Test Driven Development
- Unit Testing
- Testing Web Applications with Selenium
- Continuous Integration & Cruise Control

Joel Spolsky

The Joel Test - 12 Steps To Better Code

1. Do you use source control?
2. Can you make a build in one step?
3. Do you make daily builds?
4. Do you have a bug database?
5. Do you fix bugs before writing new code?
6. Do you have an up-to-date schedule?
7. Do you have a spec?
8. Do programmers have quiet working conditions?
9. Do you use the best tools money can buy?
10. Do you have testers?
11. Do new candidates write code during their interview?
12. Do you do hallway usability testing?

"10 or lower and you've got serious problems...
...most are running at about a 2 or 3" - Joel Spolsky

Test Driven Development

Test Driven Development

(Test First Development)

Write tests which fail, then implement the code to make it pass
- concept with roots in eXtreme Programming (XP)

'Design by contract'
The tests are your contract with the users
As long as the tests pass, the software works as specified
Fixing Bugs
Write a test which reproduces the bug, then fix the bug.
How to test?

Unit Testing

Unit Testing

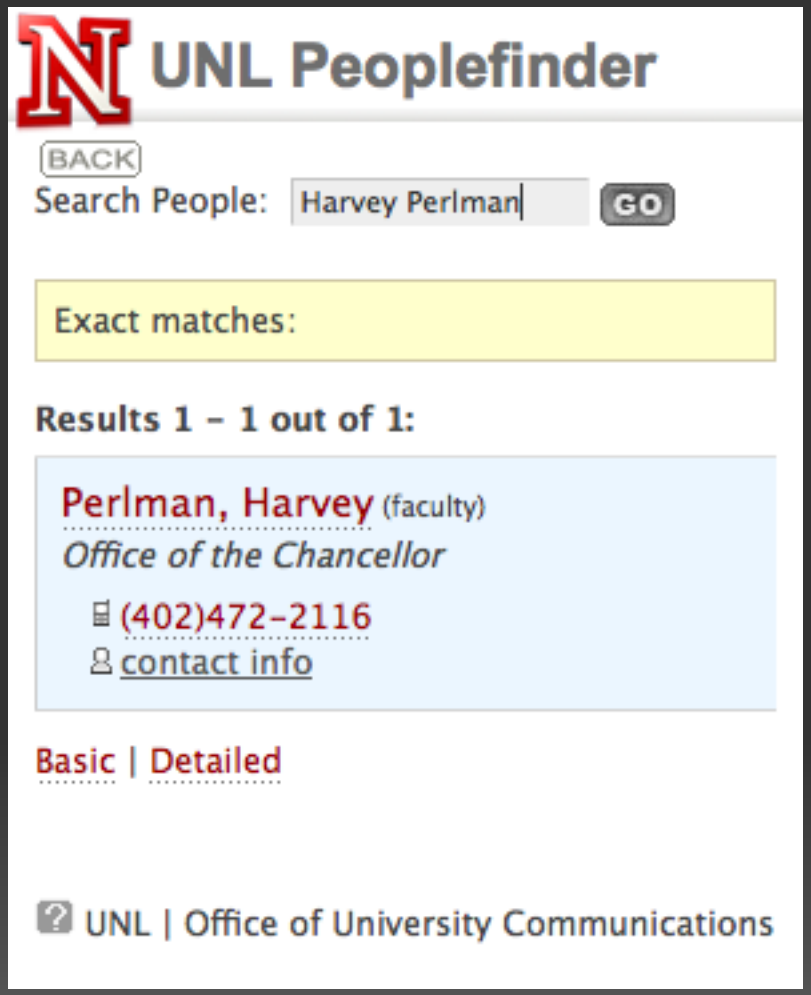
Testing Applications/Software:

Unit Testing

- Divide an application into testable parts (units)
- Combine all the parts into a suite of tests
- Make sure they run quick, and often

A 'Test' Case: UNL Peoplefinder

http://peoplefinder.unl.edu/



A simple test:

```
/**
* Test an exact match query.
*/
$peoplefinder = new Peoplefinder();
$result = $peoplefinder->getExactMatches('Harvey Perlman');
if (is_array($result)
    && sizeof($result) == 1
    && $result[0]->givenName == 'Harvey') {
    echo 'OK';
} else {
    echo 'ERROR';
}
```

Running the test gives me:

```
OK
```

Home-brew Tests

Benefits:

- Very Simple
- Series of 'OK' statements = Good
- Any 'ERROR' statements = Problem

Drawbacks:

- Only easy to implement for simple checks
- You have to write if/else statements for every check
- What if you want more than OK & ERROR?
  - Ex: Only run test B if test A is OK
  - Ex: I want to expect an ERROR in some instances
  - Ex: More helpful error messages
    - The 2+2 function returned 3, but I expected 4

Benefits of a Unit Testing Framework

Testing infrastructure is already built, just implement the unique pieces of each test.

```
/**
* Test an exact match query.
*/
$peoplefinder = new Peoplefinder();
$result = $peoplefinder->getExactMatches('Harvey Perlman');
assertTrue(is_array($result));
assertEquals(1, sizeof($result));
assertEquals('Harvey', $result[0]->givenName);
```

Running the code outputs:

```
OKOKOK
```

Unit Testing Frameworks

xUnit or Beck Testing Framework developed by Kent Beck

Kent Beck (of Extreme Programming) wrote an initial testing framework for the Smalltalk language

- Test Fixtures
- Test Suites
- Test Execution
- Assertions

Many unit testing frameworks have followed this design pattern: CPPUnit, JUnit, XUnit.net, PHPUnit etc

Exhaustive list at http://en.wikipedia.org/wiki/List_of_unit_testing_frameworks

A simple test:

```
/**

* Test an exact match query.

*/

public function testGetExactMatches() {
     $p = new Peoplefinder();

     $r = $peoplefinder->getExactMatches('Harvey Perlman');
     $this->assertTrue(is_array($r), 'An array of results was not returned.');

     $this->assertEquals(1, count($r), 'The size of the result set is not what was
     expected.');

     $this->assertEquals('Harvey', $r[0]->givenName, 'Harvey was not returned.');
}
```

Run Your Tests: Success

```
ucommbieber:~/workspace/peoplefinder bbieber$ php tests/PeoplefinderTest.php
PHPUnit 3.2.0RC2 by Sebastian Bergmann.

.I.......IIII

Time: 14 seconds


OK, but incomplete or skipped tests!
Tests: 14, Incomplete: 5.
```

Run Your Tests: Failure

```
ucommbieber:~/workspace/peoplefinder bbieber$ php tests/PeoplefinderTest.php
PHPUnit 3.2.0RC2 by Sebastian Bergmann.
.I.F.....IIII
Time: 13 seconds
There was 1 failure:
1) testGetExactMatches(PeoplefinderTest)
Failed asserting that two strings are equal.
expected string <Harve>
difference      <     ?>
got string      <Harvey>
/Users/bbieber/Documents/workspace/peoplefinder/tests/PeoplefinderTest.php:82
/Users/bbieber/Documents/workspace/peoplefinder/tests/PeoplefinderTest.php:26
/Users/bbieber/Documents/workspace/peoplefinder/tests/PeoplefinderTest.php:198

        FAILURES!
        Tests: 14, Failures: 1, Incomplete: 5.
```

Got that?

Everyone understands the concept?

Seems easy, right?

Got that?

What's wrong?

Run Your Tests: Failure

```
testGetExactMatches()
3 failures:
1) An array of results was not returned.
2) The size of the result set is not what was expected.
3) Harvey was not returned.
```

Run Your Tests: Failure

```
testGetExactMatches()
3 failures:
1) An array of results was not returned.
2) The size of the result set is not what was expected.
3) Harvey was not returned.
```

What do these error messages mean?
Do the messages tell you anything about the problem?
Are we testing the right things? At the right level?

Run Your Tests: Failure

```
testGetExactMatches()
1) An array of results was not returned.
```

Someone broke the method's return signature?

```
2) The size of the result set is not what was expected.
```

There's more than one Harvey Perlman - Harvey Jr.?
New chancellor?

```
3) Harvey was not returned.
```

He changed his name?
We lost him?

Run Your Tests: Failure

What was the cause?

LDAP directory was down.

Important Unit Testing Concepts

- Test at the smallest level. Test small pieces, not apps!
- Name your tests appropriately!
- Don't test against live data (unless you have to).
  - Use mock objects.
  - Restore the test database to a known state.
- If your tests rely on previous tests PASS or FAIL, code them appropriately.
  - Ex: If the LDAP connection is down, no point in testing how many results we get etc.

- When a test fails - don't send developers down the wrong path.

Benefits of Unit Testing

- Established a base for what our program is required to do
- We can make drastic changes and know that the program - our contract with the user - functions as expected
- Ensure new features don't break expected behaviour
  - Ensure new developers ...
  - Ensure past developers ...
  - Ensure YOU ...

Benefits of Unit Testing

- Established a base for what our program is required to do
- We can make drastic changes and know that the program - our contract with the user - functions as expected
- Ensure new features don't break expected behaviour
  - Ensure new developers ...
  - Ensure past developers ...
  - Ensure YOU ...

"Even the dumbest developer can break your code"

Code Coverage, Code Analysis

Unit Tests => Code Coverage Analysis

We test it!

- How much?
- To what detail?

"We have 100% method coverage!"

- What about decision branches within those methods?

Unit Tests => Code Coverage Analysis

We test it!

- How much?
- To what detail?

Metrics based on coverage of:

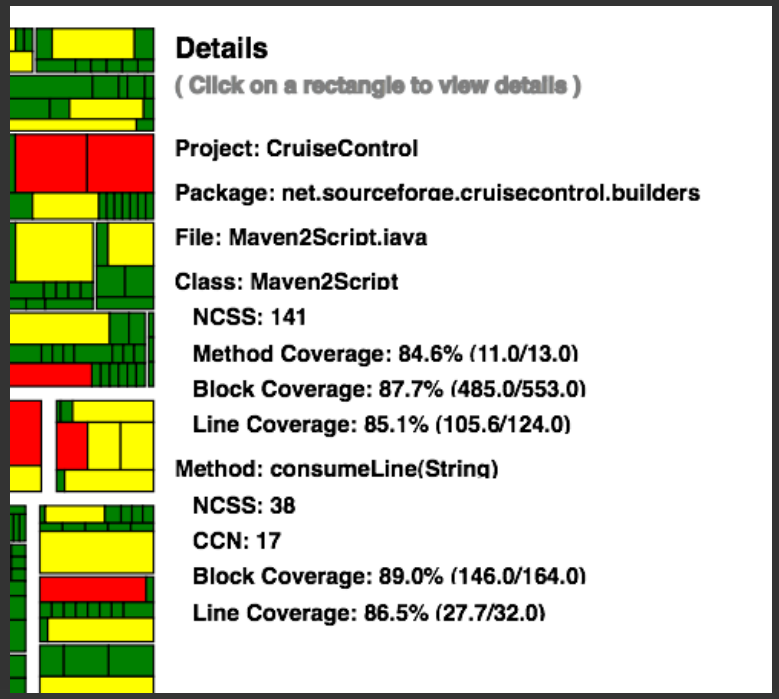
- Functions, statements, conditions, paths, entry & exit.

Confidence in the code we're using.

... we've inherited.
... we're delivering.
... asking someone else to support.

Code Coverage Analysis

Code Coverage Analysis



Panopticode code coverage for cruise control

Code Coverage Analysis

PeoplefinderTest

Current directory: **/Users/bbieber/Documents/workspace/peoplefinder**

Legend: **Low: 0% to 35%** **Medium: 35% to 70%** **High: 70% to 100%**

	Coverage								
	Classes			Methods			Lines		
Total		100.00%	2 / 2		78.95%	15 / 19		79.70%	161 / 202
Peoplefinder		100.00%	1 / 1		50.00%	1 / 2		25.00%	7 / 28
Peoplefinder.php		100.00%	1 / 1		82.35%	14 / 17		88.51%	154 / 174

Generated by PHPUnit 3.2.0RC2 and Xdebug 2.0.2 at Wed Nov 14 13:18:22 CST 2007.

Code Coverage Analysis

```
45                  :
46                  :     /**
47                  :      * Takes in a string from the LDAP directory, usually formatted like:
48                  :      *     ### ___ UNL 68588-####
49                  :      *    Where ### is the room number, ___ = Building Abbreviation, #### zip extension
50                  :      *
51                  :      * @param string
52                  :      * @return array Associative array.
53                  :      */
54                  :     function formatPostalAddress()
55                  :     {
56                  :         /* this is a faculty postal address
57                  :             Currently of the form:
58                  :             ### ___ UNL 68588-####
59                  :             Where ### is the room number, ___ = Building Abbreviation, #### zip extension
60                  :         */
61                  :         /**
62                  :          * We assumed that the address format is: ### ___ UNL 68588-####.
63                  :          * Some 'fortunate' people have addresses not in this format.
64                  :          */
65                  :         //RLIM
66                  :         // treat UNL as the delimiter for the streetaddress and zip
67                0 :         if (strpos($this->postalAddress,'UNL')) {
68                0 :             $addressComponent = explode('UNL', $this->postalAddress);
69                0 :         } else {
70                0 :             $addressComponent = explode('UNO', $this->postalAddress);
71                  :         }
72                0 :         $address['region']          = 'NE';
73                0 :         $address['street-address'] = trim($addressComponent[0]);
74                0 :         if (isset($addressComponent[1])) {
75                0 :             $address['postal-code'] = trim($addressComponent[1]);
76                0 :         } else {
77                0 :             $address['postal-code'] = '';
78                  :         }
79                0 :         switch (substr($address['postal-code'],0,5)) {
80                0 :             case '68182':
81                0 :                 $address['locality'] = 'Omaha';
82                0 :             break;
83                0 :             case '68583':
84                0 :             case '68588':
85                0 :             default:
86                0 :                 $address['locality'] = 'Lincoln';
87                0 :             break;
88                0 :         }
89                  :
90                0 :         return $address;
91                  :     }
92                  : }
```

Code Coverage Analysis

```
public function testFormatAddress() {

        $p = new Peoplefinder();

        $r = $p->getMockUID('bbieber2');

        $address = $r->formatPostalAddress();

        $this->assertEquals('a:4:{s:6:"region";s:2:"NE";s:14:"street-address";s:9:"326D ADMN";s:11:"postal-code";s:10:"
68588-0424";s:8:"locality";s:7:"Lincoln";}',

                                serialize($address));
}
```

Code Coverage Analysis

```
45                  :
46                  :     /**
47                  :      * Takes in a string from the LDAP directory, usually formatted like:
48                  :      *     ### ___ UNL 68588-####
49                  :      *    Where ### is the room number, ___ = Building Abbreviation, #### zip extension
50                  :      *
51                  :      * @param string
52                  :      * @return array Associative array.
53                  :      */
54                  :     function formatPostalAddress()
55                  :     {
56                  :         /* this is a faculty postal address
57                  :             Currently of the form:
58                  :             ### ___ UNL 68588-####
59                  :             Where ### is the room number, ___ = Building Abbreviation, #### zip extension
60                  :         */
61                  :         /**
62                  :          * We assumed that the address format is: ### ___ UNL 68588-####.
63                  :          * Some 'fortunate' people have addresses not in this format.
64                  :          */
65                  :         //RLIM
66                  :         // treat UNL as the delimiter for the streetaddress and zip
67                1 :         if (strpos($this->postalAddress,'UNL')) {
68                1 :             $addressComponent = explode('UNL', $this->postalAddress);
69                1 :         } else {
70                0 :             $addressComponent = explode('UNO', $this->postalAddress);
71                  :         }
72                1 :         $address['region']          = 'NE';
73                1 :         $address['street-address'] = trim($addressComponent[0]);
74                1 :         if (isset($addressComponent[1])) {
75                1 :             $address['postal-code'] = trim($addressComponent[1]);
76                1 :         } else {
77                0 :             $address['postal-code'] = '';
78                  :         }
79                1 :         switch (substr($address['postal-code'],0,5)) {
80                1 :             case '68182':
81                0 :                 $address['locality'] = 'Omaha';
82                0 :             break;
83                1 :             case '68583':
84                1 :             case '68588':
85                1 :             default:
86                1 :                 $address['locality'] = 'Lincoln';
87                1 :             break;
88                0 :         }
89                  :
90                1 :         return $address;
91                  :     }
92                  : }
```

Code Coverage Analysis

```
        $r = $p->getMockUID('imiller2');

        $address = $r->formatPostalAddress();

        $this->assertEquals('a:4:{s:6:"region";s:2:"NE";s:14:"street-address";s:8:"103C PKI";s:11:"postal-
code";s:10:"68182-0681";s:8:"locality";s:5:"Omaha";}',

                            serialize($address));
```

Code Coverage Analysis

```
46                  :     /**
47                  :      * Takes in a string from the LDAP directory, usually formatted like:
48                  :      *     ### ___ UNL 68588-####
49                  :      *    Where ### is the room number, ___ = Building Abbreviation, #### zip extension
50                  :      *
51                  :      * @param string
52                  :      * @return array Associative array.
53                  :      */
54                  :     function formatPostalAddress()
55                  :     {
56                  :         /* this is a faculty postal address
57                  :             Currently of the form:
58                  :             ### ___ UNL 68588-####
59                  :             Where ### is the room number, ___ = Building Abbreviation, #### zip extension
60                  :         */
61                  :         /**
62                  :          * We assumed that the address format is: ### ___ UNL 68588-####.
63                  :          * Some 'fortunate' people have addresses not in this format.
64                  :          */
65                  :         //RLIM
66                  :         // treat UNL as the delimiter for the streetaddress and zip
67                1 :         if (strpos($this->postalAddress,'UNL')) {
68                1 :             $addressComponent = explode('UNL', $this->postalAddress);
69                1 :         } else {
70                1 :             $addressComponent = explode('UNO', $this->postalAddress);
71                  :         }
72                1 :         $address['region']          = 'NE';
73                1 :         $address['street-address'] = trim($addressComponent[0]);
74                1 :         if (isset($addressComponent[1])) {
75                1 :             $address['postal-code'] = trim($addressComponent[1]);
76                1 :         } else {
77                0 :             $address['postal-code'] = '';
78                  :         }
79                1 :         switch (substr($address['postal-code'],0,5)) {
80                1 :             case '68182':
81                1 :                 $address['locality'] = 'Omaha';
82                1 :             break;
83                1 :             case '68583':
84                1 :             case '68588':
85                1 :             default:
86                1 :                 $address['locality'] = 'Lincoln';
87                1 :             break;
88                0 :         }
89                  :
90                1 :         return $address;
91                  :     }
92                  : }
```

Code Coverage Analysis

PeoplefinderTest

Current directory: /Users/bbieber/Documents/workspace/peoplefinder

Legend: Low: 0% to 35% | Medium: 35% to 70% | High: 70% to 100%

	Coverage								
	Classes			Methods			Lines		
Total		100.00%	2 / 2		84.21%	16 / 19		89.11%	180 / 202
Peoplefinder		100.00%	1 / 1		100.00%	2 / 2		92.86%	26 / 28
Peoplefinder.php		100.00%	1 / 1		82.35%	14 / 17		88.51%	154 / 174

Generated by PHPUnit 3.2.0RC2 and Xdebug 2.0.2 at Wed Nov 14 13:38:30 CST 2007.

Got that?

Everyone understands the concepts?

Seems easy, right?

Got that?

"We have 100% code coverage!
And all tests pass!"

Got that?

"We have 100% code coverage!
And all tests pass!"

Oh really.
Did you test all input possibilities (infinite)?

Did you test on the client machine?

What About Web Applications?

Testing Web Applications

We know our code works on the (testing) server, but web applications have 'special needs' - right?

Bug Details:

- What Browser?
- Which Version?
- What OS?

The server sends the right HTML, but what if your designer hides that with Javascript or CSS?

How can we test in this environment?
Test the code, then hand it over to a human to test the browser?

Browser Based Testing with Selenium

Selenium from OpenQA - http://www.openqa.org/

Selenium allows automating browser actions to test web applications

Platforms:

- Windows
- Mac OS X
- Linux

Browsers:

- Internet Explorer
- Firefox
- Safari
- etc

Browser Based Testing with Selenium

Easily record and playback actions
Automate your browser and make assertions
Can integrate into your global test suite

Selenium IDE extension for Firefox

Browser Based Testing with Selenium

UNL Peoplefinder

Harvey S Perlman (faculty)

Chancellor
University of Nebraska–Lincoln
Office of the Chancellor

Work:
201 ADMS
Lincoln, NE 68588-0419
USA

Work:
(402)472-2116

hperlman1@unl.edu
Delivery Address: hperlman@unlnotes.unl.edu

BACK

UNL | Office of University Communications

Selenium IDE *

Base URL http://peoplefinder.unl.edu/

Run Walk Step

Table Source

Command	Target	Value
open	/	
type	q	Harvey Perlman
clickAndWait	submitbutton	
verifyTextPresent	Perlman, Harvey	
clickAndWait	link=Perlman, Harvey	
verifyTextPresent	Chancellor	
verifyElementPresent	link=hperlman1@unl....	

Command

Target Find

Value

Log Reference Info Clear

Browser Based Testing with Selenium

Selenium IDE *

Base URL http://peoplefinder.unl.edu/

Run Walk Step

Table Source

Command	Target	Value
open	/	
type	q	Harvey Perlman
clickAndWait	submitbutton	
verifyTextPresent	Perlman, Harvey	
clickAndWait	link=Perlman, Harvey	
verifyTextPresent	Chancellor	
verifyElementPresent	link=hperlman1@unl....	

Command

Target Find

Value

Log Reference Info Clear

[info] element doesn't have fireEvent
[info] element has initMouseEvent
[info] Executing: |verifyTextPresent | Chancellor | |
[info] Executing: |verifyElementPresent | link=hperlman1@unl.edu | |

Browser Based Testing with Selenium

Firefox File Edit Options Window Help

New Test
Open... ⌘O
Save Test ⌘S
Save Test As...
Export Test As...
Close ⌘W

HTML
Java - Selenium RC
C# - Selenium RC
Perl - Selenium RC
PHP - Selenium RC
Python - Selenium RC
Ruby - Selenium RC
Ruby - Selenium IDE 0.7 (deprecated)

PeoplefinderTest

PHP - peoplefinder/tests/PeoplefinderTest.php - Eclipse Platform - /Users/bbieber/Documents/w

UNL | Peoplefinder

Base URL http://peoplefi

Run Walk Step

Table Source

Command	Target	Value
open	/	
type	q	Harvey Perlman
clickAndWait	submitbutton	
verifyTextPresent	Perlman, Harvey	
clickAndWait	link=Perlman, Harvey	
verifyTextPresent	Chancellor	
verifyElementPresent	link=hperlman1@unl....	

Command
Target
Value
Find

Log Reference Info Clear

[info] element doesn't have fireEvent
[info] element has initMouseEvent
[info] Executing: |verifyTextPresent | Chancellor | |
[info] Executing: |verifyElementPresent | link=hperlman1@unl.edu | |

Browser Based Testing with Selenium

```
20    function testMyTestCase()
21    {
22        $this->selenium->open("/");
23        $this->selenium->type("q", "Harvey Perlman");
24        $this->selenium->click("submitbutton");
25        $this->selenium->waitForPageToLoad("30000");
26        try {
27            $this->assertTrue($this->selenium->isTextPresent("Perlman, Harvey"));
28        } catch (PHPUnit_Framework_AssertionFailedError $e) {
29            array_push($this->verificationErrors, $e->toString());
30        }
31        $this->selenium->click("link=Perlman, Harvey");
32        $this->selenium->waitForPageToLoad("30000");
33        try {
34            $this->assertTrue($this->selenium->isTextPresent("Chancellor"));
35        } catch (PHPUnit_Framework_AssertionFailedError $e) {
36            array_push($this->verificationErrors, $e->toString());
37        }
38        try {
39            $this->assertTrue($this->selenium->isElementPresent("link=hperlman1@unl.edu"));
40        } catch (PHPUnit_Framework_AssertionFailedError $e) {
41            array_push($this->verificationErrors, $e->toString());
42        }
43
44    }
```

Browser Based Testing with Selenium

Selenium Remote Control allows you to remotely connect to a testing server and issue commands

Integrates into your existing testing framework

Download Selenium RC and the Server and start it up:

```
ucommbieber:~/$ java -jar selenium-server.jar
16:23:39.199 INFO - Java: "Apple Computer, Inc." 1.5.0_07-87
16:23:39.247 INFO - OS: Mac OS X 10.4.10 i386
16:23:39.249 INFO - v0.9.2 [2006], with Core v0.8.3 [1879]
16:23:39.689 INFO - Version Jetty/5.1.x
16:23:39.690 INFO - Started HttpContext[/,/]
16:23:39.691 INFO - Started HttpContext[/selenium-server,/selenium-server]
16:23:39.691 INFO - Started HttpContext[/selenium-server/driver,/selenium-server/driver]
16:23:39.771 INFO - Started SocketListener on 0.0.0.0:4444
16:23:39.771 INFO - Started org.mortbay.jetty.Server@1ddcde
```

Browser Based Testing with Selenium

Run your exported test:

```
ucommbieber:~/workspace/peoplefinder/tests bbieber$ phpunit BrowserTest
PHPUnit 3.2.0RC2 by Sebastian Bergmann.
.
Time: 7 seconds
OK (1 test)
```

Browser Based Testing with Selenium

Firefox File Edit View History Bookmarks Tools Window Help

Selenium Remote Control v0.9.2 [2006], with Core v0.8.3 [1879]

http://peoplefinder.unl.edu/selenium-server/core/RemoteRunner.html?se:

Google

Getting Started Latest Headlines

Se 34 selenium 78.96

Selenium Functional Testing for Web Apps

Open Source From ThoughtWorks, Inc and Friends

Slow Mode:

Tools

View DOM | Show Log

106716

Last Four Test Commands:

isTextPresent(Perlman, Harvey)
click(link=Perlman, Harvey)
waitForPageToLoad(30000)
isTextPresent(Chancellor)

UNL Peoplefinder

Harvey S Perlman (faculty)

Chancellor
University of Nebraska-Lincoln
Office of the Chancellor

Work:
201 ADMS
Lincoln, NE 68588-0419
USA

Work:
(402)472-2116

hperlman1@unl.edu

ve ress: hperlman@unlnotes.unl.edu

ess/chizelno/

Terminal — php — ttyp1 — ⌘1

RC is invalid: Permission denied

wserTest.php:32

null

it BrowserTest

RC is invalid: Permission denied

wserTest.php:32

Browser Based Testing with Selenium

Selenium video.

Other Use Cases for Selenium

Log in to a system, perform some actions, verify results - this is how we test the UNL Events system

Automate accessibility checks -

- Go to CynthiaSays.com
- Enter web address
- Click submit
- Check for '**Passed Automated Verification**'
- .. wait one minute ..
- Repeat with next web address

Test that the HTML, CSS, and Javascript all work together in your supported browsers

Got that?

Everyone understands the concepts?

Seems easy, right?

*"Have a **continuous integration** solution in place.*
Really.
*If you don't, you just burn money by writing **tests**.*

*I would go so far and say, if you don't have **continuous integration**, you should stop writing unit **tests** and do click testing."* - Lars Strojny

I disagree to some extent -

- Tests can be the foundation for CI (cart before horse)
- Tests can serve as secondary documentation
- CI can serve another purpose besides just unit testing - api documentation, coding standards tests

"CI without unit tests can have benefits. At a minimum, you've defined a one-step build process... that's a huge benefit." - Chuck Burgess

Continuous Integration

Continuous Integration

What is it?

Martin Fowler - http://martinfowler.com/articles/continuousIntegration.html

Developers 'continuously' (more frequently) integrate their code

In a team of developers, each developer is committing changes to the source code repository daily

Commit, commit, commit

Continuous Integration

Why?

Ensure that all developers are working with the latest code - working or not

Other developers need to know immediately if an interface or method signatures have changed

Everyone can see that progress is being made

A 5pm, leaving for the day commit of 500 lines of code is a real pain for other developers, and a danger to the build

Cruise Control http://cruisecontrol.sourceforge.net/

A framework for a continuous build/integration process

Overview:

Tell cruise control about your project's build process

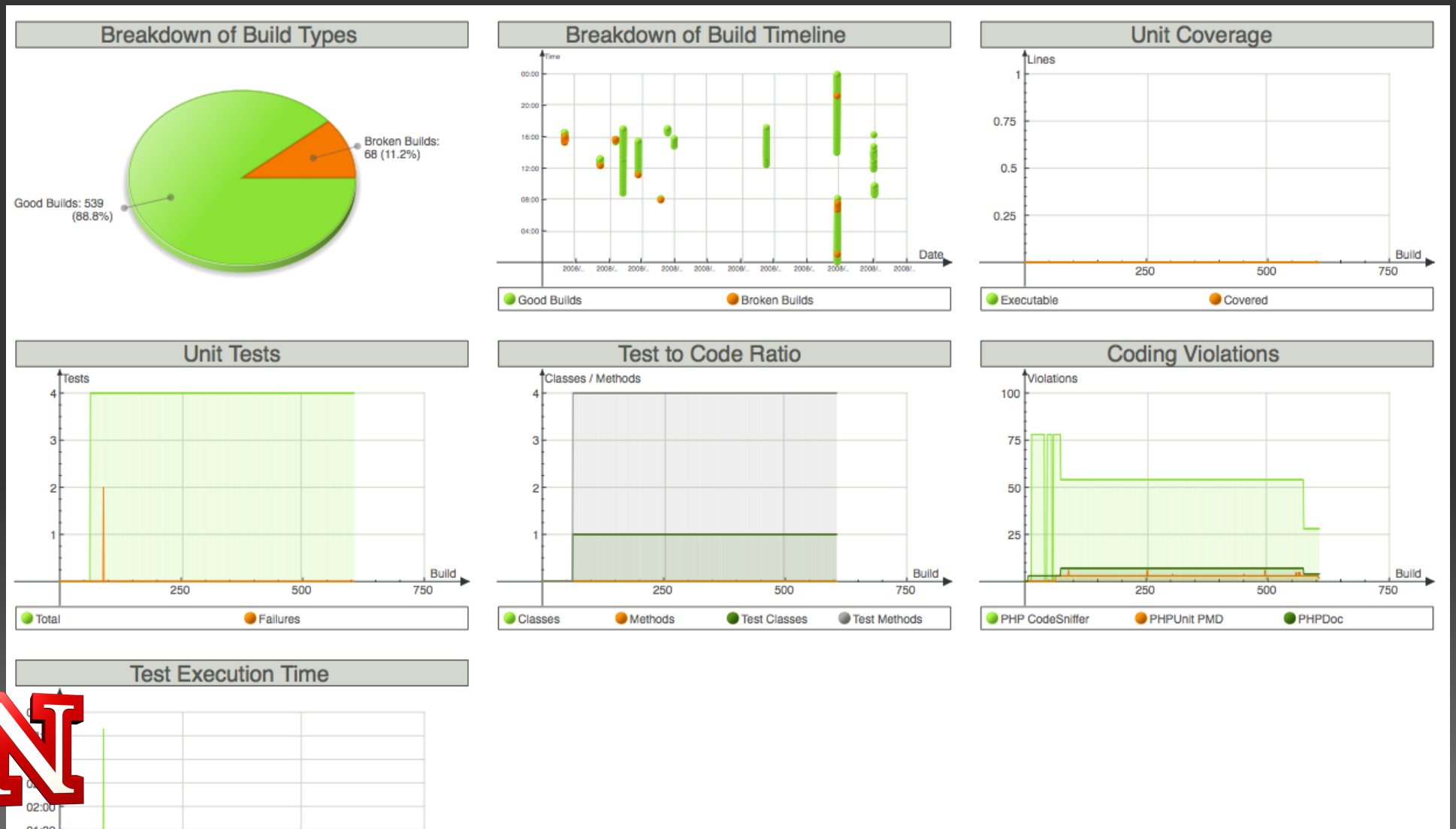
- Ant, NANT
- GNU Make
- MSBuild
- custom

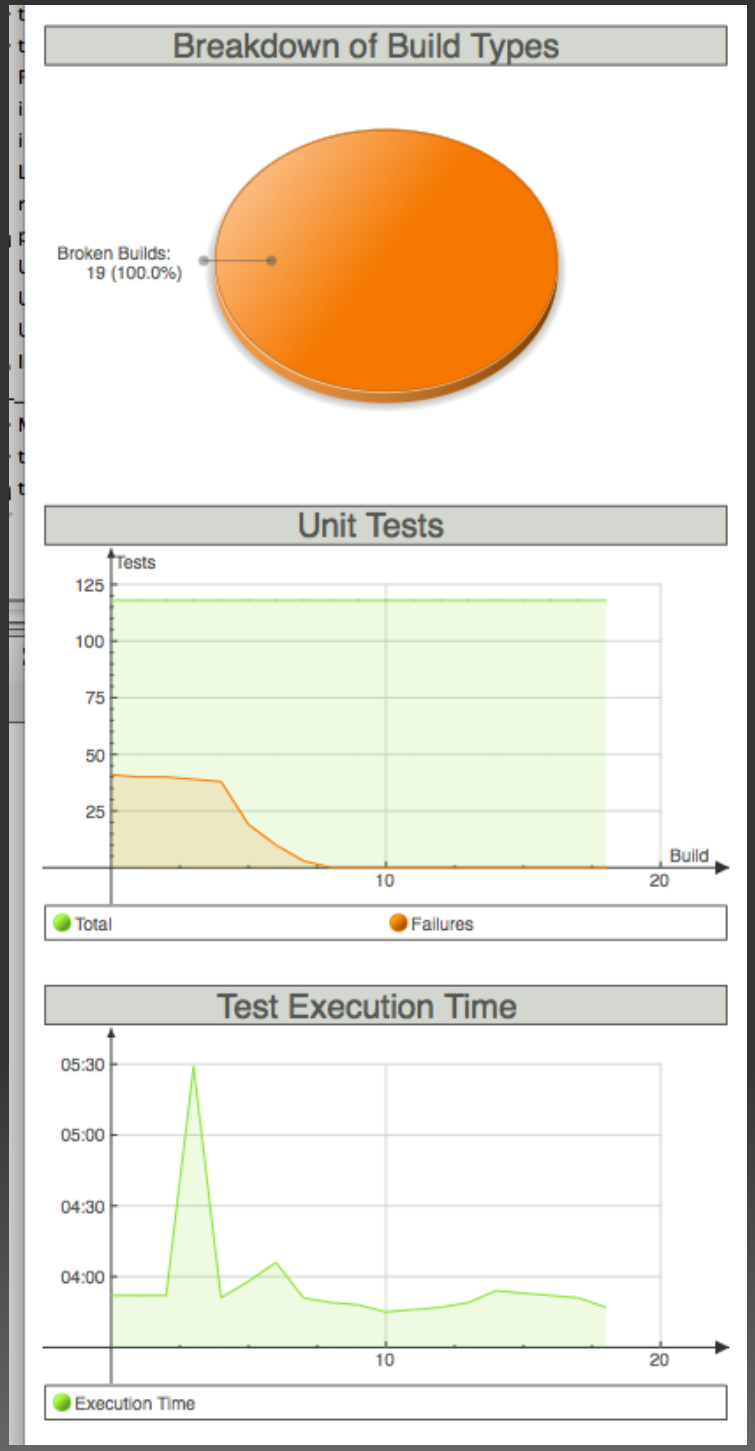
Define dashboard and notification with plugins/modules

- Notify developers when they've broken the build
- RSS, IM, Email etc

Cruise Control http://cruisecontrol.sourceforge.net/

The dashboard allows developers and managers to see the project's current status.




Breakdown of Build Types
Broken Builds:
19 (100.0%)
Unit Tests
Tests
125
100
75
50
25
10
20
Build
Total
Failures
Test Execution Time
05:30
05:00
04:30
04:00
10
20
Execution Time

Joel Spolsky

The Joel Test - 12 Steps To Better Code

1. Do you use source control?
2. Can you make a build in one step?
3. Do you make daily builds?
4. Do you have a bug database?
5. Do you fix bugs before writing new code?
6. Do you have an up-to-date schedule?
7. Do you have a spec?
8. Do programmers have quiet working conditions?
9. Do you use the best tools money can buy?
10. Do you have testers?
11. Do new candidates write code during their interview?
12. Do you do hallway usability testing?

'10 or lower and you've got serious problems ... most are running at about a 2 or 3'

Joel Spolsky

The Joel Test - 12 Steps To Better Code

1. Do you use source control?
2. Can you make a build in one step?
3. Do you make daily builds?
4. Do you have a bug database?
5. Do you fix bugs before writing new code?
6. Do you have an up-to-date schedule?
7. Do you have a spec?
8. Do programmers have quiet working conditions?
9. Do you use the best tools money can buy?
10. Do you have testers?
11. Do new candidates write code during their interview?
12. Do you do hallway usability testing?

By following a test driven development and continuous integration process you can knock off about 8 of these.

Summary: Empirical Evidence

On the Effectiveness of Test-first Approach to Programming
Erdogmus, H. Proceedings of the IEEE Transactions on Software Engineering, 31(1). January 2005.

- Improves productivity
- Better understanding of requirements
- Reduces scope of individual tasks
- Reduces debugging and rework effort

Discussion & Questions

Brett Bieber
brett.bieber@gmail.com

Resources

http://www.extremeprogramming.org/
http://en.wikipedia.org/wiki/Unit_testing
http://www.xprogramming.com/testfram.htm
http://en.wikipedia.org/wiki/List_of_unit_testing_frameworks
http://www.openqa.org/selenium/
http://www.phpunit.de/
http://www.martinfowler.com/articles/continuousIntegration.html
http://cruisecontrol.sourceforge.net/
http://www.joelonsoftware.com/

Blog plug:
http://saltybeagle.com/