

WRITE TESTS

NO, SERIOUSLY: WRITE TESTS

WEB EDITION

CLARKE BRUNSDON

- I write code because I have to.
- Writing code for 15 years professionally*

*completely terribly for more than half of them



STEAM BOLT

- Started as FreeRunning Technologies in 2008
- Current Staffing Breakdown
 - 1 Office Manager
 - 1 UI/UX designer
 - 24 Developers
- Maintain "Solidus"
- Not in fact called "Steamboat"

MY GOAL

MY BEING HERE MADE YOU TRY WRITING
MORE TESTS

THE AGENDA

1. What
2. Why
3. "Why Not"
4. How
5. Testing for the Web
6. Odds and Ends

WHAT THIS PRESENTATION ISN'T

1. Good enough

PART 1: WHAT

"I WAS A COMP SCI STUDENT HERE TOO" EDITION

- More code that tests your applications code
- Describes how your code runs and how it responds
- Should be 1-10x larger than your actual codebase
- The most valuable code in your codebase

**TESTS ARE THE MOST
VALUABLE CODE IN
YOUR CODEBASE**

TYPES OF TESTING

- Unit Tests
- Feature/Integration Tests
- Regression Tests
- Behavior Tests
- Load/Stress Tests

PART 2: WHY

IT MAKES THE SYSTEM BETTER!...

- It reduces defects in prod...
- Adds documentation on behavior
- Good guidelines for new developers

... BUT REALLY WE DO IT FOR OURSELVES

- Good tests make it faster and easier to develop
- Pushing confidently and frequently is fun
- Bugs and defects are stressful
- Tests make you think about how your code is used

... AND WE DO IT FOR OUR CO-WORKERS

- It might be someone else that deals with your bullshit

*** DISCLAIMER ***

YOU'RE GOING TO BE TERRIBLE AT IT

PART 3: "WHY NOT"

1. It takes longer to develop and write tests
2. This area of code is untestable
3. "I'm not allowed to"
4. This code isn't supposed to exist for long

IT TAKES LONGER TO DEVELOP
AND WRITE TESTS

A.K.A. "I'm Bad"

THIS AREA OF CODE IS
UNTESTABLE

A.K.A. "My code is bad"

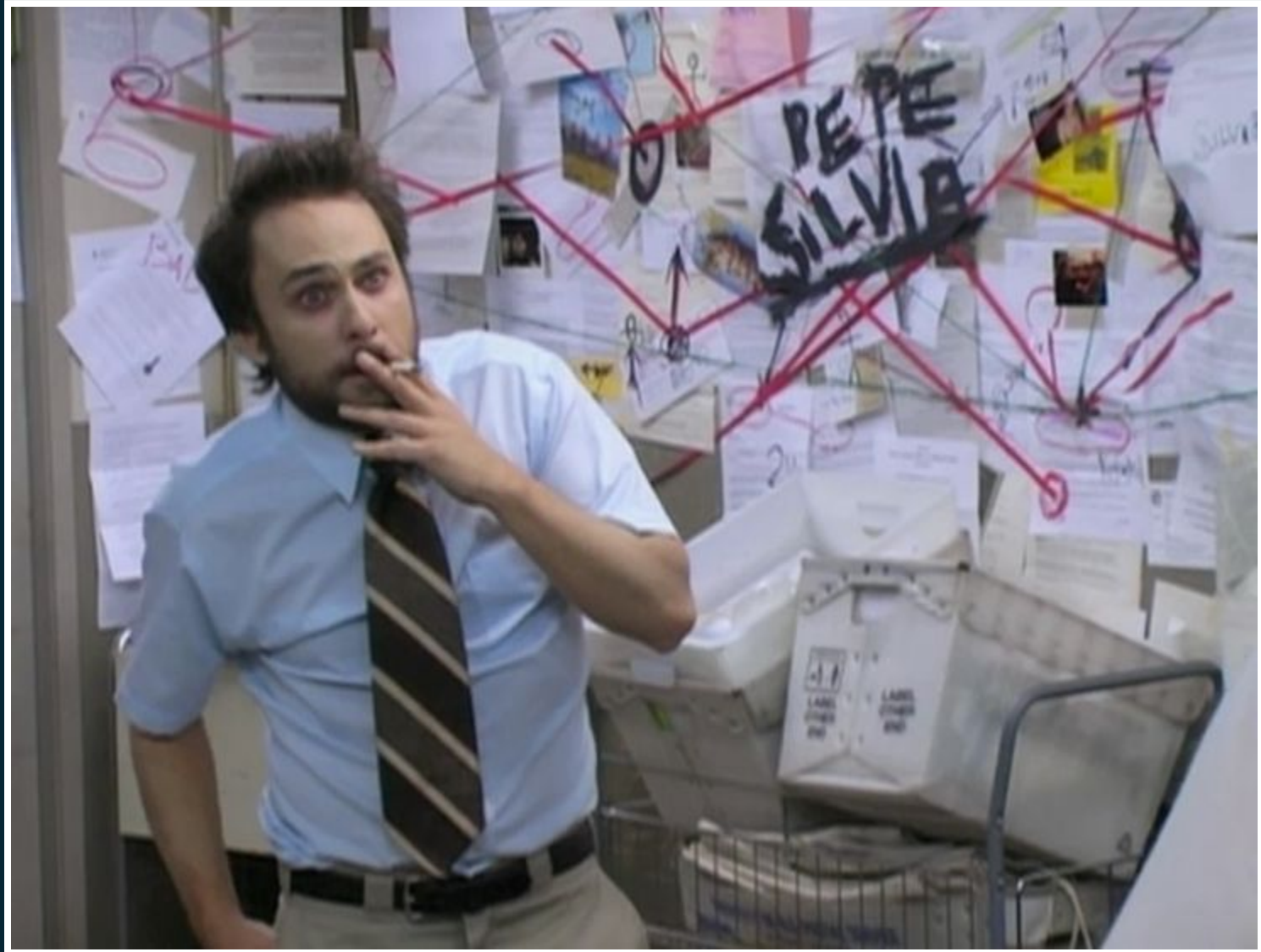
"I'M NOT ALLOWED TO"

A.K.A. "My company is bad"

THIS CODE ISN'T SUPPOSED TO
EXIST FOR LONG

A.K.A. "My understanding on life is bad"

INTERMISSION



PART 4: HOW

QUICK EXAMPLE: PALINDROMES

```
import math

def is_palindrome(word):
    for i in range(0, math.floor( len(word) / 2) ):
        if word[i] != word[-i]:
            return False
    return True
```

```
def assert_equal(a, b):
    if a != b:
        raise Exception("{0} does not equal {1}".format(a, b))
```

WRITE CODE TO TEST CODE

```
assert_equal( True, is_palindrome("racecar") )
```

```
assert_equal( False, is_palindrome('rocky racoon') )
```

```
assert_equal( True, is_palindrome('') )
```

```
assert_equal( True, is_palindrome('a') )
```

```
assert_equal( ??, is_palindrome('racecar ') )
```


SERIOUS (MOSTLY OUT OF SCOPE) DISCUSSION

- Test Public API
- Dont Test Private API
- Hard to test Highly Coupled code
- Easy to test Hihgly Decoupled code

PART 5: TESTING FOR THE WEB

Unit Testing

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation.

UNIT TESTING JAVASCRIPT - MOCHA

```
var assert = require('assert');
describe('Array', function() {
  describe('#indexOf()', function() {
    it('should return -1 when the value is not present', function() {
      assert.equal(-1, [1,2,3].indexOf(4));
    });
  });
});
```

Courtesy of: <https://mochajs.org/>

... BUT HOW OFTEN ARE PEOPLE
WRITING ARRAY PRIMITIVES IN
JAVASCRIPT?

```
upload: function(file) {
  if (!this.tests.formdata) return;

  var progressModel = new ProgressModel({file: file});
  progressModel.previewFile();
  progressModel.uploadFile();

  var progressView = new ProgressView({model: progressModel});
  this.progressZone.appendChild(progressView.render().el);
},

onDrop: function(e) {
  this.el.className = '';
  e.preventDefault();

  for (var i = 0; i < e.originalEvent.dataTransfer.files.length; i++)
    this.upload(e.originalEvent.dataTransfer.files[i]);
},
```

<https://github.com/solidusio/solidus/pull/1553/files>

FRAMEWORK SPECIFIC TESTING (ANGULAR/PROTRACTOR)

```
upload: function(file) {
describe('angularjs homepage todo list', function() {
  it('should add a todo', function() {
    browser.get('https://angularjs.org');

    element(by.model('todoList.todoText')).sendKeys('write first protractor test');
    element(by.css('[value="add"]')).click();

    var todoList = element.all(by.repeater('todo in todoList.todos'));
    expect(todoList.count()).toEqual(3);
    expect(todoList.get(2).getText()).toEqual('write first protractor test');

    // You wrote your first test, cross it off the list
    todoList.get(2).element(by.css('input')).click();
    var completedAmount = element.all(by.css('.done-true'));
    expect(completedAmount.count()).toEqual(2);
  });
});
```

CREATE VALUABLE TESTS

- Protect against bugs
- Adds Clarity
- Low False Positives

REAL FAKE ~~DOORS~~ BROWSERS

- PhantomJS
- Selenium/Webdrivers

Webdrivers are currently a dumpster fire. Evolving into a real spec.

INTEGRATION/FEATURE TESTING (RUBY/CAPYBARA)

```
it 'allows you to empty the cart', js: true do
  create(:product, name: "RoR Mug")
  visit spree.root_path
  click_link "RoR Mug"
  click_button "add-to-cart-button"

  expect(page).to have_content("RoR Mug")
  click_on "Empty Cart"
  expect(page).to have_content("Your cart is empty")

  within "#link-to-cart" do
    expect(page).to have_content("EMPTY")
  end
end
```

END-TO-END TESTING CONSIDERATIONS

- Speed
- Not a silver bullet
- Markup Independence
- Waiting Matchers

JAVASCRIPT FEATURE FRAMEWORKS


NIGHTWATCH, INTERN, ETC.


```
module.exports = {  
  'Demo test Google' : function (browser) {  
    browser  
      .url('http://www.google.com')  
      .waitForElementVisible('body', 1000)  
      .setValue('input[type=text]', 'nightwatch')  
      .waitForElementVisible('button[name=btnG]', 1000)  
      .click('button[name=btnG]')  
      .pause(1000)  
      .assert.containsText('#main', 'Night Watch')  
      .end();  
  }  
};
```

Courtesy of <http://nightwatchjs.org/>

PART 6: ODDS AND ENDS


CONTINUOUS: INTEGRATION







All checks have passed
2 successful checks


[Hide all checks](#)




 **codeclimate** — Code Climate didn't find any new or fixed issues.

[Details](#)



 **semaphoreci** — The build passed on Semaphore.

[Details](#)



This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request

▼

or view [command line instructions](#).

CODE COVERAGE

All Files (96.9% covered at 149.22 hits/line)

689 files in total. 26960 relevant lines. 26123 lines covered and 837 lines missed

Search:

| File | % covered | Lines | Relevant L |
|---|-----------|-------|------------|
| Q spec/support/test_gateway.rb | 100.0 % | 2 | 1 |
| Q spec/support/dummy_ability.rb | 100.0 % | 3 | 2 |
| Q spec/support/concerns/working_factories.rb | 100.0 % | 9 | 5 |
| Q spec/support/concerns/default_price.rb | 100.0 % | 34 | 19 |
| Q spec/support/big_decimal.rb | 100.0 % | 5 | 3 |
| Q spec/models/spree/zone_spec.rb | 100.0 % | 464 | 260 |
| Q spec/models/spree/variant_spec.rb | 100.0 % | 715 | 379 |
| Q spec/models/spree/variant_property_rule_value_spec.rb | 100.0 % | 18 | 10 |
| Q spec/models/spree/variant_property_rule_spec.rb | 100.0 % | 83 | 45 |
| Q spec/models/spree/variant_property_rule_condition_spec.rb | 100.0 % | 15 | 8 |
| Q spec/models/spree/variant/scopes_spec.rb | 100.0 % | 78 | 47 |

Test Driven development
Write tests, THEN code!

BEHAVIOR DRIVEN DEVELOPMENT

```
Scenario: start game  
  Given I am not yet playing  
  When I start a new game  
  Then the game should say "Welcome to CodeBreaker"  
  And the game should say "Enter guess:"
```

<https://blog.engineyard.com/2009/cucumber-introduction>

STATISTICS!

100% of developers write test code.

100% of people pretending to be developers write no test code.

THANKS!

(I WAS CLARKE BRUNSDON)

- <https://stembolt.com>
- clarke@stembolt.com
- careers@stembolt.com