

A Testing Anti-Pattern Safari



WWTLD



What Would @tenderlove Do?



Aja Hammerly

@thagomizer_rb

<http://github.com/thagomizer>

<http://www.thagomizer.com>

SUBSTANTIAL



Effective Tests

Trustworthy

Simple

Overview

- Pointless Tests
- Wasted Time / Effort
- False Positives & Negatives
- Inefficient Tests

Pointless tests

No Tests





Connect

Michael (Doc) Norton @DocOnDev

26 Mar

"We're coding too fast to write tests." is the same as
"We're driving too fast to confirm we're on the right road."

[Details](#)

Write Tests

Not Running Tests



Use CI

Listen To Your Tests

Wasted Effort/Time

Testing OPC



```
class OrderTest < ActiveSupport::TestCase

  def test_find_order_by_id
    order = Order.create!

    o = Order.find(order.id)

    assert_equal order, o
  end

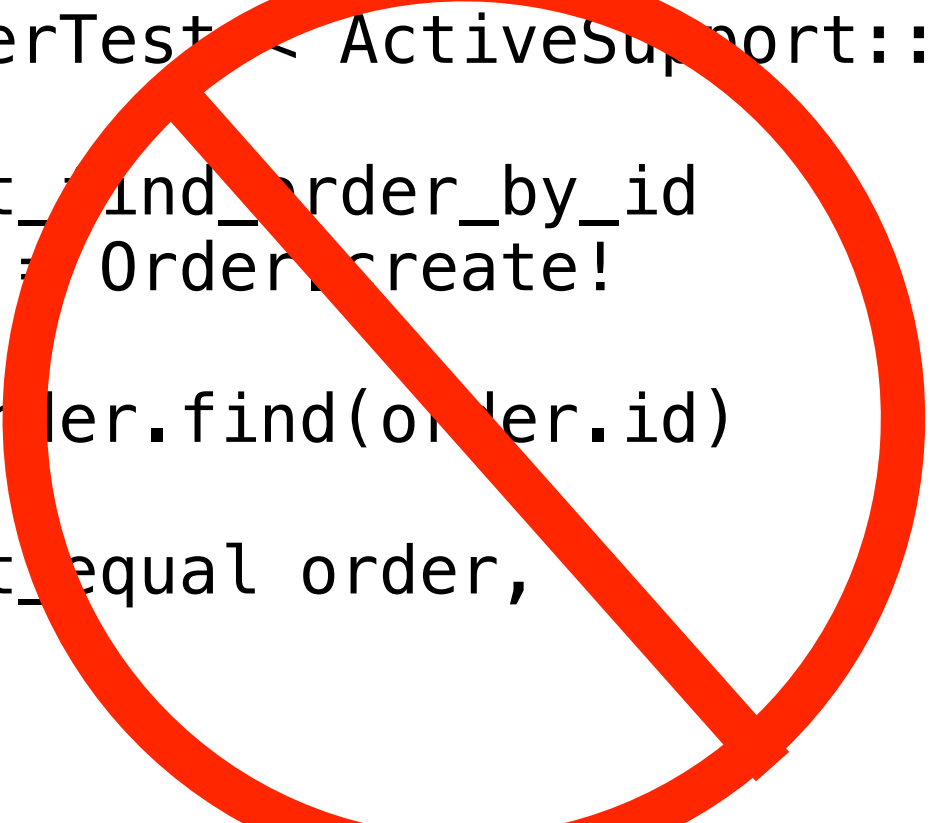
end
```

**Test What You
Provide**

```
class OrderTest < ActiveSupport::TestCase
  def test_find_order_by_id
    order = Order.create!

    o = Order.find(order.id)

    assert_equal order,
  end
end
```



`assert_nothing_raised`



```
def raise_something
  raise "W00T"
end

class Fails < Test::Unit::TestCase

  def test_assert_nothing_raised
    assert_nothing_raised do
      raise_something
    end
  end

end
```

```
# Running tests:
```

```
F
```

```
  1) Failure:
```

```
test_assert_nothing_raised(TestClass)
```

```
[assert_nothing_raised.rb:9]:
```

```
Exception raised:
```

```
<#<RuntimeError: WOOT>>.
```

```
1 tests, 1 assertions, 1 failures, 0 errors
```

Leave It Out

```
def raise_something  
  raise "WOOT"  
end
```

```
class Fails < Test::Unit::TestCase  
  
  def test_does_not_raise  
  
    raise_something  
  
  end  
  
end
```

```
# Running tests:
```

```
E
```

```
  1) Error:
```

```
test_does_not_raise(TestClass):
```

```
RuntimeError: WOOT
```

```
    assert_nothing_raised.rb:4:in  
    `raise_something'  
    assert_nothing_raised.rb:15:in  
    `test_does_not_raise'
```

```
1 tests, 0 assertions, 0 failures, 1 errors
```


False Positives & Negatives

Time Sensitive



```
require 'active_support/time'

class Fails < Test::Unit::TestCase

  def test_add_hours
    t1 = Time.now
    t2 = t1 + 4.hours
    assert_equal t1.day, t2.day
  end

end
```



80%
Real
Code!

Stub Time.now()

```
require 'active_support/time'

class Test < Test::Unit::TestCase
  def test_add_hours
    Time.stubs(:now) =
      Time.new(2012, 1, 1)
    t1 = Time.now
    t2 = t + 4.hours
    assert_equal t1.day, t2.day
  end
end
```

```
class Promo
  def initialize
    @expiration = Time.new(2012, 12, 12)
  end

  def expired?
    Time.now > @expiration
  end
end

class Fails < Test::Unit::TestCase
  def test_promo_expiration
    p = Promo.new
    refute p.expired?
  end
end
```

Use relative dates


```
class Promo
  def initialize(expiration =
                Time.new(2012, 12, 12))
    @expiration = expiration
  end
  def expired?; Time.now > @expiration; end
end

class Test < Test::Unit::TestCase
  def test_promo_expiration
    p = Promo.new(3.months.from_now)
    refute p.expired?
  end
end
```

Order Dependent



```
class AnimalTest < Test::Unit::TestCase

  def test_create_record
    a = Animal.create!(name:"Lion")
    assert_not_nil a
  end

  def test_find_record
    a = Animal.find_by_name("Lion")
    assert_not_nil a
  end

end
```

```
$ ruby test_animal.rb -v
```

```
AnimalTest#test_create_record .
```

```
AnimalTest#test_find_record .
```

```
2 tests, 2 assertions, 0 failures
```

```
class OrderTest < Test::Unit::TestCase
  def test_create_record
    a = Animal.create!(name:"Lion")
    assert a
  end

  def test_find_record
    a = Animal.find_by_name("Lion")
    assert a
  end

  def test_delete_record
    a = Animal.find_by_name("Lion")
    a.delete
  end
end
```

```
$ ruby test_animal.rb -v
```

```
AnimalTest#test_create_record .  
AnimalTest#test_delete_record .  
AnimalTest#test_find_record F
```

```
2 tests, 2 assertions, 1 failures
```

Randomize Your Tests

Default Run Orders

Framework	Default Order
Test/Unit	Alphabetical
Rspec	As In File
MiniTest	Random

How To Randomize

Framework	Randomize
Test/Unit (1.9+)	<pre>def self.test_order :random end</pre>
Rspec	<code>--order rand</code>
MiniTest	Default!

Bonus Fun: Minitest/Hell

```
require 'minitest/hell'
```

```
class MyTests < Test::Unit::TestCase  
  # Tests  
end
```

Tests That Can't Fail



```
class User < ActiveRecord::Base
  validates_presence_of :name
end
```

```
describe User do
  it "all sorted by name" do
    b = User.create(name: "Ben")
    n = User.create(name: "Nick")

    User.all.should == [b, n]
  end
end
```



```
$ rspec spec_user.rb
```

```
▪
```

```
Finished in 0.00127 seconds  
1 examples, 0 failures
```

Red \Rightarrow Green

```
describe User do
  it "all sorted by name" do
    n = User.create(name: "Nick")
    b = User.create(name: "Ben")

    User.all.should == [b, n]
  end
end
```

Inefficient Tests

Require External Resources



```
class Clever
  def districts
    url = "#{host}/districts"
    key = "DEMO_KEY"
    `curl -u #{key}: #{url}`
  end
end
```

```
class Clever
  def districts
    url = "/districts"
    key = "DEMO_KEY"
    `curl -u #{key}: #{host}#{url}`
  end
end

class External < Test::Unit::TestCase
  def test_get_clever_data
    c = Clever.new
    assert_include c.districts, 'Demo'
  end
end
```



```
$ ruby test_external_resources.rb
```

```
# Running tests:
```

```
▪
```

```
Finished tests in 0.243850s...
```

```
1 tests, 2 assertions...
```

Mock/Stub External Resources

```
class Clever
  def districts
    url = "/districts"
    key = "DEMO_KEY"
    `curl -u #{key}: #{host}#{url}`
  end
end
```

```
class Requester
  def self.make_request(url, key)
    `curl -u #{key}: #{host}#{url}`
  end
end

class Clever
  def districts
    url = "/districts"
    key = "DEMO_KEY"
    Requester.make_request(url, key)
  end
end
```

```
$ ruby test_external_resources.rb
```

```
▪
```

```
Finished tests in 0.526345s...
```

```
1 tests, 2 assertions...
```

```
class Requester
  def self.make_request(url, key)
    `curl -u #{key}: #{host}#{url}`
  end
end
```

```
class StubRequester < Requester
  $responses = {}

  def self.make_request(url, _)
    return $responses[url]
  end
end
```

```
class Clever
  def initialize(r = Requester)
    @r = r
  end

  def districts
    url = "/districts"
    key = "DEMO_KEY"
    @r.make_request(url, key)
  end
end
```

```
$ ruby test_external_resources.rb
```

```
▪
```

```
Finished tests in 0.526345s...
```

```
1 tests, 2 assertions...
```



```
class External < Test::Unit::TestCase
  def test_get_clever_data
    c = Clever.new(StubRequester)
    assert_include c.districts, 'Demo'
  end
end
```

```
$ ruby test_external_resources.rb
```

```
F
```

```
Finished tests in 0.000980s...
```

```
1) Failure:
```

```
test_get_clever_data(External)
```

```
Expected nil (NilClass) to respond to  
#include?.
```

```
1 tests, 1 assertions...
```

```
class StubRequester < Requester
  $responses = {}

  def self.make_request(url, key)
    if $responses[url] then
      return $responses[url]
    else
      raise url + super(url, key)
    end
  end
end
```

```
$ ruby test_external_resources.rb
```

```
E
```

```
Finished tests in 0.201016s...
```

```
1) Error:
```

```
test_get_clever_data(External):
```

```
RuntimeError: /districts <response>  
  <stacktrace>
```

```
1 tests, 0 assertions...
```

```
class StubRequester < Requester
  $responses = {
    "/districts" => <response>}

  def self.make_request(url, key)
    if $responses[url] then
      return $responses[url]
    else
      raise url + super(url, key)
    end
  end
end
end
```

```
$ ruby test_external_resources.rb
```

```
▪
```

```
Finished tests in 0.000401s...
```

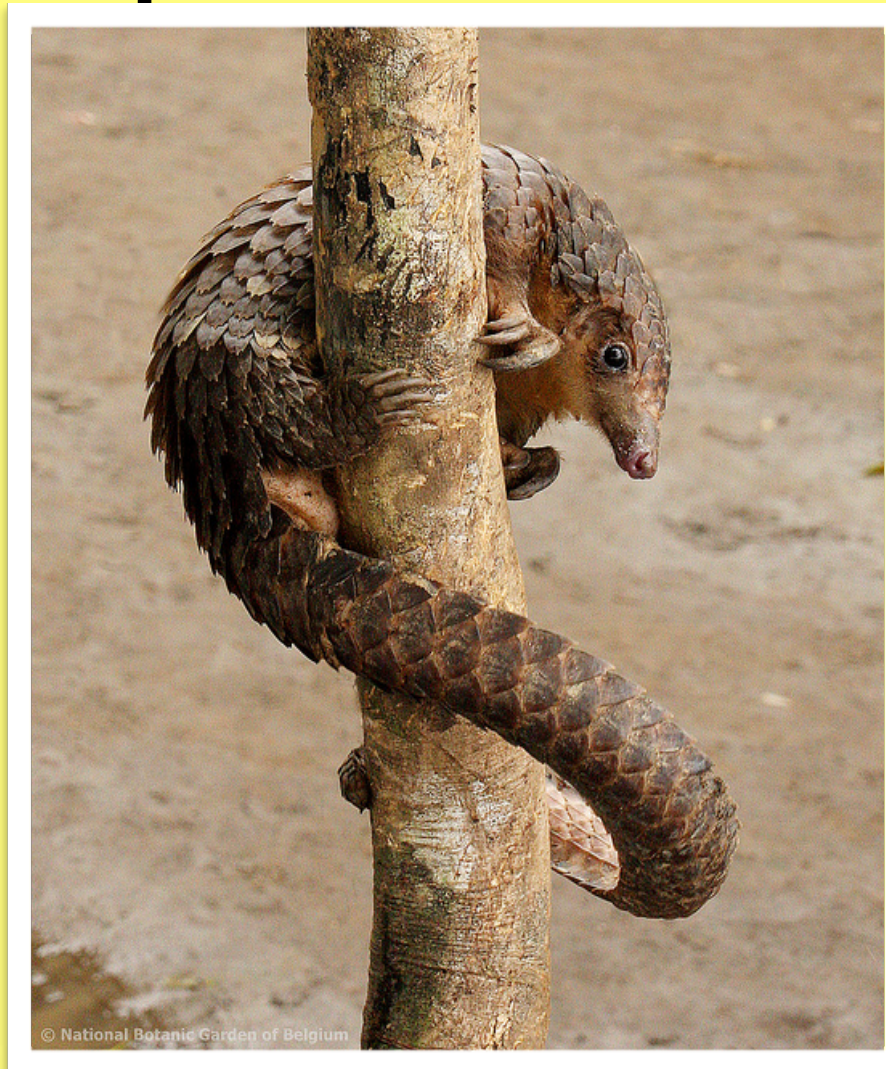
```
1 tests, 2 assertions...
```

0.243850s

VS

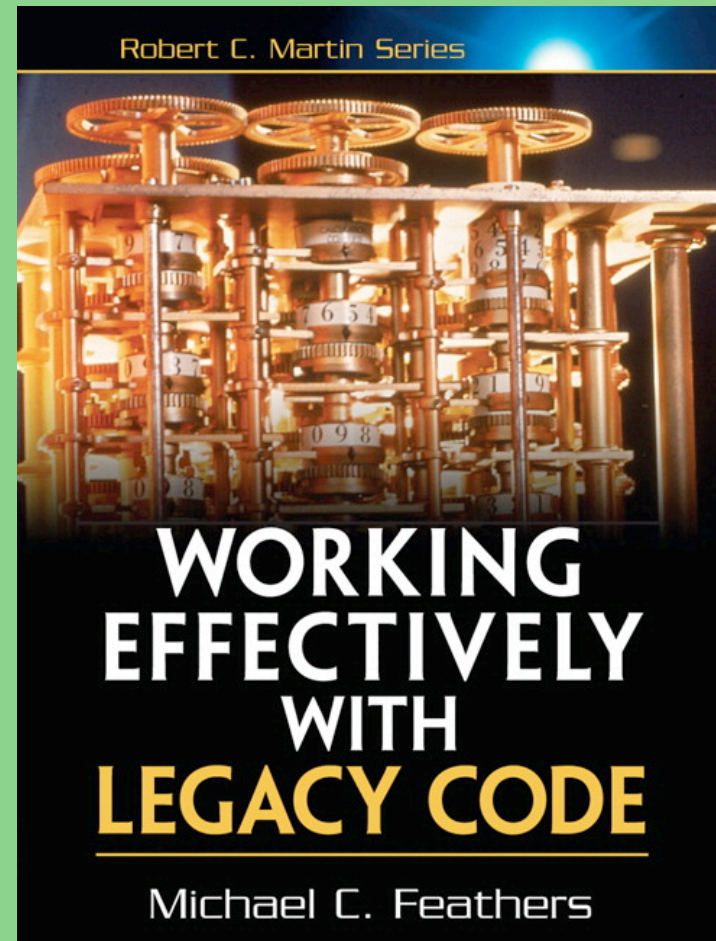
0.000401s

Complicated Setup



Refactor Implementation

**Working
Effectively with
Legacy Code**
by Michael Feathers



Messy Tests



Messy Tests

- Repeated Code
- Copy, Paste, Tweak
- Disorganized Tests
- Literals everywhere

Refactor

Test Refactoring

- DRY up your tests
- Group by method under test
- Use descriptive names
- Put literals in variables

Slow tests



```
class SlowTest < Test::Unit::TestCase
  def test_fast
    assert true
  end

  def test_slow
    MyClass.slow_method
    assert true
  end
end
```


**Fix your
implementation**

Use Guards

```
class SlowTest < Test::Unit::TestCase
  def test_fast
    assert true
  end

  def test_slow
    MyClass.slow_method
    assert true
  end
end
```

```
class SlowTest < Test::Unit::TestCase
  def test_fast
    assert true
  end

  if ENV["ALL"]
    def test_slow
      MyClass.slow_method
      assert true
    end
  end
end
```

```
$ ruby slow_tests.rb  
# Running tests:
```

```
▪
```

```
Finished tests in 0.000364s,  
2747.2527 tests/s, 2747.2527  
assertions/s.
```

```
1 tests, 1 assertions, 0 failures, 0  
errors, 0 skips
```

```
$ ALL=1 ruby slow_tests.rb  
# Running tests:
```

```
..
```

```
Finished tests in 5.002102s, 0.3998  
tests/s, 0.3998 assertions/s.
```

```
2 tests, 2 assertions, 0 failures, 0  
errors, 0 skips
```

My Top 3

- Not Running Tests
- Order Dependent Tests
- Require External Resources

Thank You



Photo Credits

- Afternoon Get-together by Mara1 (<http://www.flickr.com/photos/flametree/3512147657/>)
- Kenyan Buffalo! by Mara1 (<http://www.flickr.com/photos/flametree/3606728565/>)
- Sleepy Cheetah Wide Shot by nicpic (<http://www.flickr.com/photos/nicpic/344462908/>)
- Crowned Crane! by Mara1 (<http://www.flickr.com/photos/flametree/3620753809/>)
- Safari RamatGan-003 by Flavio~ (<http://www.flickr.com/photos/37873897@N06/7246890214/>)
- dancing giraffes by xavi talleda (<http://www.flickr.com/photos/xavitalleda/4278335979/>)
- Grazing Zebra by Mara1 (<http://www.flickr.com/photos/flametree/3105239216/>)
- Hippo peek ! by Mara1 (<http://www.flickr.com/photos/flametree/3341717929/>)
- Masai Mara lion ! by Mara1 (<http://www.flickr.com/photos/flametree/3930600351/>)
- Poledancing for Darwin by zimbart (<http://www.flickr.com/photos/zimbart/5771021272/>)
- img_8577 by The.Rohit (http://www.flickr.com/photos/rohit_saxena/5496394591/)
- tortise by abrin523 (<http://www.flickr.com/photos/abrin/997535662/>)