

How to Fail With 100% Test Coverage

jason@thinkrelevance.com

What Kind of
Coverage are You?

lining up

```
module Math
  def max(a, b)
    if a > b then a else b end
  end
end

require "test/unit"
require "math"

class MathTest < Test::Unit::TestCase
  include Math

  def test_max
    assert_equal 2, max(2, 1)
  end
end
```

Code reported as executed by Ruby looks like this...

and this: this line is also marked as covered.

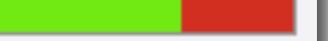
Lines considered as run by rcov, but not reported by Ruby, look like this,
and this: these lines were inferred by rcov (using simple heuristics).

Finally, here's a line marked as not executed.

Name	Total lines	Lines of code	Total coverage	Code coverage
math.rb	5	5	100.0% 	100.0% 

```
1 module Math
2   def max(a, b)
3     if a > b then a else b end
4   end
5 end
```

```
module Math
def max(a, b)
  if a > b then
    a
  else
    b
  end
end
end
```

Name	Total lines	Lines of code	Total coverage	Code coverage
math.rb	9	9	66.7% 	66.7% 
<pre>1 module Math 2 def max(a, b) 3 if a > b then 4 a 5 else 6 b 7 end 8 end 9 end</pre>				

```
require "test/unit"
require "math"

class MathTest < Test::Unit::TestCase
  include Math

  def test_first_parameter_as_max
    assert_equal 2, max(2, 1)
  end

  def test_second_parameter_as_max
    assert_equal 2, max(1, 2)
  end
end
```

Name	Total lines	Lines of code	Total coverage	Code coverage
math.rb	9	9	100.0%	<div style="width: 100%; background-color: #2eac21;"></div> 100.0%

```
1 module Math
2   def max(a, b)
3     if a > b then
4       a
5     else
6       b
7     end
8   end
9 end
```

branching out

```
// Math.java
public class Math {
    public static int max(int a, int b) {
        return a > b ? a : b;
    }
}
```

```
// MathTest.java
import junit.framework.TestCase;

public class MathTest extends TestCase {
    public void testMax() {
        assertEquals(2, Math.max(2, 1));
    }
}
```

Classes in this File	Line Coverage	Branch Coverage	Complexity
Math	100%  2/2	50%  1/2	1
<pre>1 1 public class Math { 2 public static int max(int a, int b) { 3 1 return a > b ? a : b; 4 } 5 }</pre>			

```
import junit.framework.TestCase;

public class MathTest extends TestCase {
    public void testFirstParameterAsMax() {
        assertEquals(2, Math.max(2, 1));
    }

    public void testSecondParameterAsMax() {
        assertEquals(2, Math.max(1, 2));
    }
}
```

Classes in this File	Line Coverage	Branch Coverage	Complexity
Math	100%  2/2	100%  2/2	1
<pre>1 1 public class Math { 2 public static int max(int a, int b) { 3 return a > b ? a : b; 4 } 5 }</pre>			

7 Flavors of Fail

Underspecification



“While it's possible to specify a product ambiguously, it is not possible to build a product ambiguously.”

Tom DeMarco and Timothy Lister

```
# code under test
#####
MIN_FREE_SHIPPING_PRICE = 25.0

def free_shipping?(total_order_price)
  total_order_price > MIN_FREE_SHIPPING_PRICE
end
```

```
# code under test
#####
MIN_FREE_SHIPPING_PRICE = 25.0

def free_shipping?(total_order_price)
  total_order_price > MIN_FREE_SHIPPING_PRICE
end

# test suite?
#####
def test_free_shipping_returns_true_for_order_above_min_price
  assert free_shipping?(MIN_FREE_SHIPPING_PRICE + 1)
end

def test_free_shipping_returns_false_for_order_below_min_price
  assert !free_shipping?(MIN_FREE_SHIPPING_PRICE - 1)
end
```

what about my
\$25.00 order!

```
# code under test
#####
MIN_FREE_SHIPPING_PRICE = 25.0

def free_shipping?(total_order_price)
  total_order_price >= MIN_FREE_SHIPPING_PRICE
end

# test suite
#####
def test_free_shipping_returns_true_for_order_above_min_price
  assert free_shipping?(MIN_FREE_SHIPPING_PRICE + 1)
end

def test_free_shipping_returns_false_for_order_below_min_price
  assert !free_shipping?(MIN_FREE_SHIPPING_PRICE - 1)
end

def test_free_shipping_returns_true_for_order_equal_to_min_price
  assert free_shipping?(MIN_FREE_SHIPPING_PRICE)
end
```

edge cases matter

“Code coverage can only tell you how much of your code is being tested. It cannot tell you how much code you still need to write.”

Eric Sink

Incidental Coverage

```
class ProductsController < ApplicationController
  def index
    @products = Product.find(:all)

    respond_to do |format|
      format.html # index.html.erb
      format.xml { render :xml => @products }
    end
  end
end
```

```
class ProductsControllerTest < ActionController::TestCase
  def test_should_get_index
    get :index
  end
end
```

Name	Total lines	Lines of code	Total coverage	Code coverage
app/controllers/products_controller.rb	12	9	100.0% 	100.0% 
<pre> 1 class ProductsController < ApplicationController 2 # GET /products 3 # GET /products.xml 4 def index 5 @products = Product.find(:all) 6 7 respond_to do format 8 format.html # index.html.erb 9 format.xml { render :xml => @products } 10 end 11 end 12 end </pre>				

failure to assert

testing != System.out.println()

```
class ProductsControllerTest < ActionController::TestCase
  def test_should_get_index
    get :index
    assert_response :success
    assert_not_nil assigns(:products)
  end
end
```

100% covered

50% tested

```
def index
  @products = Product.find(:all)

  respond_to do |format|
    format.html # index.html.erb
    format.xml { render :xml => @products }
  end
end
```

```
def index
  @products = Product.find(:all)

  respond_to do |format|
    format.html # index.html.erb
    format.xml { raise :fatal_error }
  end
end
```

Still no failing tests?

```
def test_should_get_index_formatted_for_html
  get :index
  assert_response :success
  assert_not_nil assigns(:products)
end

def test_should_get_index_formatted_for_xml
  @request.env['HTTP_ACCEPT'] = 'application/xml'
  get :index
  assert_response :success
  assert_not_nil assigns(:products)
end
```

Invisible Code



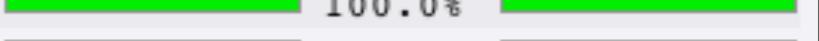
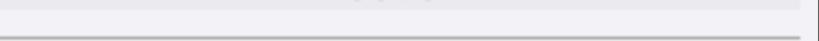
MVC PUBLIC Service Announcement

Today's Episode:
"KEEPING VIEWS STUPID"

```
[store]$ tree
...
|-- app
|   |-- controllers
|   |   |-- application.rb
|   |   `-- products_controller.rb
|   |-- helpers
|   |   |-- application_helper.rb
|   |   `-- products_helper.rb
|   |-- models
|   |   `-- product.rb
|   '-- views
|       |-- layouts
|       |   '-- products.html.erb
|       '-- products
|           |-- edit.html.erb
|           |-- index.html.erb
|           |-- new.html.erb
|           `-- show.html.erb
|-- config
|   |-- boot.rb
|   |-- database.yml
|   |-- environment.rb
|   |-- environments
|   |   ...
|   '-- initializers
|       |-- inflections.rb
|       |-- mime_types.rb
|       `-- new_rails_defaults.rb
`-- routes.rb

|-- db
|   |-- migrate
|   |   |-- 20080810220638_create_products.rb
|   |   `-- schema.rb
...
|-- lib
|   |-- product_ftp_importer.rb
|   '-- tasks
|       |-- data_load.rake
...
|-- test
|   |-- fixtures
|   |   '-- products.yml
|   |-- functional
|   |   '-- products_controller_test.rb
|   |-- integration
|   |   '-- test_helper.rb
|   '-- unit
|       '-- product_test.rb
...
37 directories, 63 files
```

```
[store]$ tree
...
|-- app
|   |-- controllers
|   |   |-- application.rb
|   |   `-- products_controller.rb
|   |-- helpers
|   |   |-- application_helper.rb
|   |   `-- products_helper.rb
|   |-- models
|   |   `-- product.rb
|   '-- views
|       |-- layouts
|       |   `-- products.html.erb
|       '-- products
|           |-- edit.html.erb
|           |-- index.html.erb
|           |-- new.html.erb
|           `-- show.html.erb
|-- config
|   |-- boot.rb
|   |-- database.yml
|   |-- environment.rb
|   |-- environments
|   |   ...
|   |-- initializers
|   |   |-- inflections.rb
|   |   |-- mime_types.rb
|   |   `-- new_rails_defaults.rb
|   '-- routes.rb
|
|   |-- db
|   |   |-- migrate
|   |   |   `-- 20080810220638_create_products.rb
|   |   '-- schema.rb
|
|   ...
|   |-- lib
|   |   |-- product_ftp_importer.rb
|   |   '-- tasks
|   |       |-- data_load.rake
|   |
|   ...
|   |-- test
|   |   |-- fixtures
|   |   |   `-- products.yml
|   |   |-- functional
|   |   |   `-- products_controller_test.rb
|   |   |-- integration
|   |   |-- test_helper.rb
|   |   '-- unit
|   |       `-- product_test.rb
|
|   ...
37 directories, 63 files
```

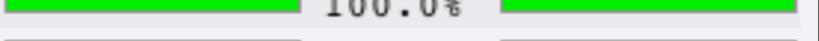
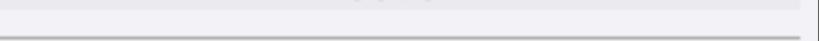
Name	Total lines	Lines of code	Total coverage	Code coverage
TOTAL	107	62	100.0%	 100.0%
app/controllers/application.rb	15	4	100.0%	 100.0%
app/controllers/products_controller.rb	85	52	100.0%	 100.0%
app/helpers/application_helper.rb	3	2	100.0%	 100.0%
app/helpers/products_helper.rb	2	2	100.0%	 100.0%
app/models/product.rb	2	2	100.0%	 100.0%

Generated using the [rcov code coverage analysis tool for Ruby](#) version 0.8.1.2.

```
[store]$ tree
...
|-- app
|   |-- controllers
|   |   |-- application.rb
|   |   `-- products_controller.rb
|   |-- helpers
|   |   |-- application_helper.rb
|   |   `-- products_helper.rb
|   |-- models
|   |   `-- product.rb
|   '-- views
|       |-- layouts
|       |   `-- products.html.erb
|       '-- products
|           |-- edit.html.erb
|           |-- index.html.erb
|           |-- new.html.erb
|           `-- show.html.erb
|-- config
|   |-- boot.rb
|   |-- database.yml
|   |-- environment.rb
|   |-- environments
|   |   ...
|   |-- initializers
|   |   |-- inflections.rb
|   |   |-- mime_types.rb
|   |   `-- new_rails_defaults.rb
|   '-- routes.rb
|
|   |-- db
|   |   |-- migrate
|   |   |   `-- 20080810220638_create_products.rb
|   |   '-- schema.rb
|
|   ...
|   |-- lib
|   |   |-- product_ftp_importer.rb
|   |   '-- tasks
|   |       |-- data_load.rake
|   |
|   ...
|   |-- test
|   |   |-- fixtures
|   |   |   `-- products.yml
|   |   |-- functional
|   |   |   `-- products_controller_test.rb
|   |   |-- integration
|   |   |-- test_helper.rb
|   |   '-- unit
|   |       `-- product_test.rb
|
|   ...
37 directories, 63 files
```

```
[store]$ tree
```

```
...
|-- app
|   |-- controllers
|   |   |-- application.rb
|   |   `-- products_controller.rb
|   |-- helpers
|   |   |-- app
|   |   |   `-- products_helper.rb
|   |-- models
|   |   `-- product.rb
|   '-- views
|       |-- layouts
|       |   `-- products.html.erb
|       '-- products
|           |-- edit.html.erb
|           |-- index.html.erb
|           |-- new.html.erb
|           `-- show.html.erb
|-- config
|   |-- boot.rb
|   |-- database.yml
|   |-- environment.rb
|   |-- environments
|   |   ...
|   |-- initializers
|   |   |-- inflections.rb
|   |   |-- mime_types.rb
|   |   `-- new_rails_defaults.rb
|   '-- routes.rb
|
|   |-- db
|   |   |-- migrate
|   |   |   `-- 20080810220638_create_products.rb
|   |   '-- schema.rb
|   '-- lib
|       |-- product_ftp_importer.rb
|       '-- tasks
|           |-- data_load.rake
|           ...
|           '-- test
|               |-- fixtures
|               |   `-- products.yml
|               |-- functional
|               |   `-- products_controller_test.rb
|               |-- integration
|               |-- test_helper.rb
|               '-- unit
|                   `-- product_test.rb
|                   ...
37 directories, 63 files
```

Name	Total lines	Lines of code	Total coverage	Code coverage
TOTAL	107	62	100.0%	 100.0%
app/controllers/application.rb	15	4	100.0%	 100.0%
app/controllers/products_controller.rb	85	52	100.0%	 100.0%
app/helpers/application_helper.rb	3	2	100.0%	 100.0%
app/helpers/products_helper.rb	2	2	100.0%	 100.0%
app/models/product.rb	2	2	100.0%	 100.0%

Generated using the [rcov code coverage analysis tool for Ruby](#) version 0.8.1.2.

```
require File.dirname(__FILE__) + "../lib/product_ftp_importer"
```

Name	Total lines	Lines of code	Total coverage	Code coverage
TOTAL	152	107	71.7%  	59.8%  
app/controllers/application.rb	15	4	100.0%  	100.0%  
app/controllers/products_controller.rb	85	52	100.0%  	100.0%  
app/helpers/application_helper.rb	3	2	100.0%  	100.0%  
app/helpers/products_helper.rb	2	2	100.0%  	100.0%  
app/models/product.rb	2	2	100.0%  	100.0%  
lib/product_ftp_importer.rb	45	45	4.4%  	4.4%  

require :all

have some class

Overspecification

```
class ProductsControllerTest < ActionController::TestCase
  def test_something
    product = Product.create(:name => "Frisbee", :price => 5.00)
    get :show, :id => product.id
    assert_response :success
    product = assigns(:product)
    assert_not_nil product
    assert product.valid?
    assert product.name == "Frisbee"
    assert product.price == 5.00
  end
end
```

```
class ProductsController < ApplicationController
  def show
    @product = Product.find(params[:id])
  end
end

class ProductsControllerTest < ActionController::TestCase
  def test_something
    product = Product.create(:name => "Frisbee", :price => 5.00)
    get :show, :id => product.id
    assert_response :success
    product = assigns(:product)
    assert_not_nil product
    assert product.valid?
    assert product.name == "Frisbee"
    assert product.price == 5.00
  end
end
```

```
class ProductsController < ApplicationController
  def show
    @product = Product.find(params[:id])
  end
end
```

```
class ProductsControllerTest < ActionController::TestCase
  def test_should_show_product
    product = create_product
    get :show, :id => product.id
    assert_response :success
    assert_equal product, assigns(:product)
  end
end
```

communicate essence

The Ugly Mirror

LUC OGIJ - ILLUS

```
# code under test
#####
User = Struct.new(:first_name, :last_name, :email) do
  def to_s
    "#{last_name}, #{first_name} <#{email}>"
  end
end
```

```
# code under test
#####
User = Struct.new(:first_name, :last_name, :email) do
  def to_s
    "#{last_name}, #{first_name} <#{email}>"
  end
end

# test suite?
#####
def test_to_s_includes_name_and_email
  user = User.new("John", "Smith", "jsmith@example.com")
  assert_equal "#{user.last_name}, #{user.first_name} <#{user.email}>",
    user.to_s
end
```

```
# code under test
#####
User = Struct.new(:first_name, :last_name, :email) do
  def to_s
    "#{last_name}, #{first_name} <#{email}>"
  end
end

# test suite
#####
def test_to_s_includes_name_and_email
  user = User.new("John", "Smith", "jsmith@example.com")
  assert_equal "#{user.last_name}, #{user.first_name} <#{user.email}>", user.to_s
  assert_equal "Smith, John <jsmith@example.com>", user.to_s
end
```

expect literals

Slow Tests



Shallow Tests



**"When users pass over your app
in favor of something
easier or simpler to use, that's the
ultimate unit test failure."**

Jeff Atwood



“[Coverage tools] are only helpful if they’re used to enhance thought, not *replace* it.”

Brian Marick

Blog: jasonrudolph.com/blog

Relevance: thinkrelevance.com

Slides: jasonrudolph.com/downloads

This presentation is published under the Creative Commons Attribution Noncommercial Share Alike License Version 3.0.
(Please see <http://creativecommons.org/licenses/by-nc-sa/3.0> for complete details.)



credits

Audio Recording

Matthew Bass

matthewbass.com

Raleigh-Area Ruby Brigade

raleighrb.com

RailsEnvy Video

railsenvy.com/2008/6/3/mvc-videos

Photos

flickr.com/photos/abuzavi/92491999

flickr.com/photos/cindy47452/441316645

flickr.com/photos/mexicaliblues/470957684

flickr.com/photos/jasmic/279741827

Blog: jasonrudolph.com/blog

Relevance: thinkrelevance.com

Slides: jasonrudolph.com/downloads

This presentation is published under the Creative Commons Attribution Noncommercial Share Alike License Version 3.0.
(Please see <http://creativecommons.org/licenses/by-nc-sa/3.0> for complete details.)

