Joining up the dots

SYNTHESIS

Stuart Caborn
George Malamidis

“Synthesis”

the formation of something complex or coherent by combining simpler things

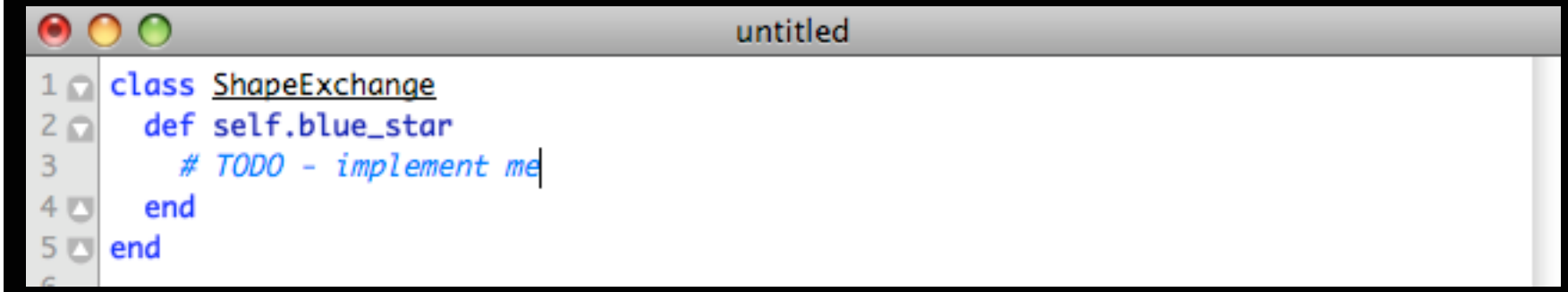
Synthesized testing

Combine lightweight tests to build confidence that our system is complete and reduce the need for large, overarching tests

Test code is code, too

Technical debt

A story of shapes



Simple?

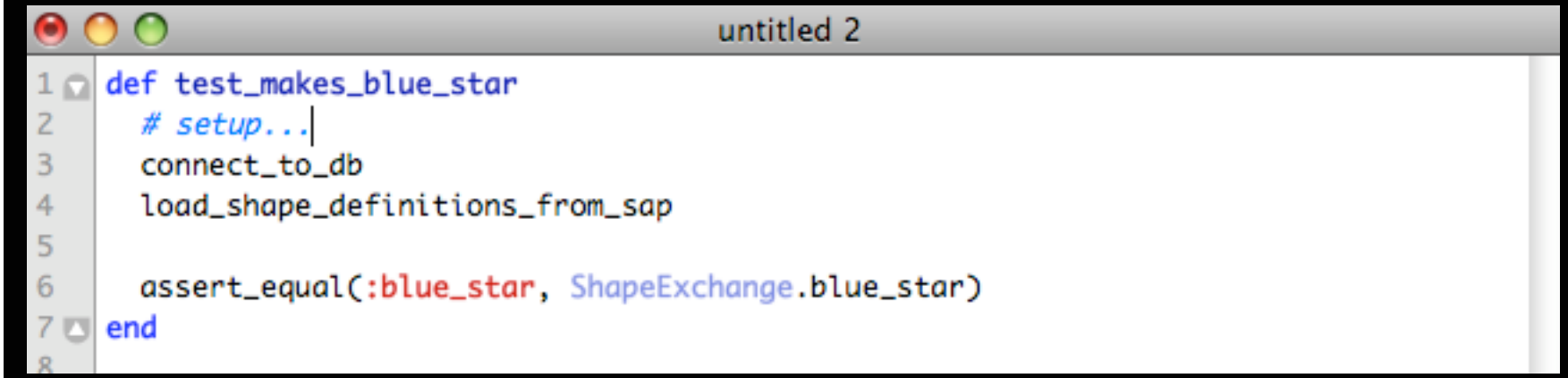
Hang on…

untitled 2

```
1 def test_makes_blue_star
2   # setup...
3   assert_equal(:blue_star, ShapeExchange.blue_star)
4 end
5
```

Just a little setup

Just a little more setup



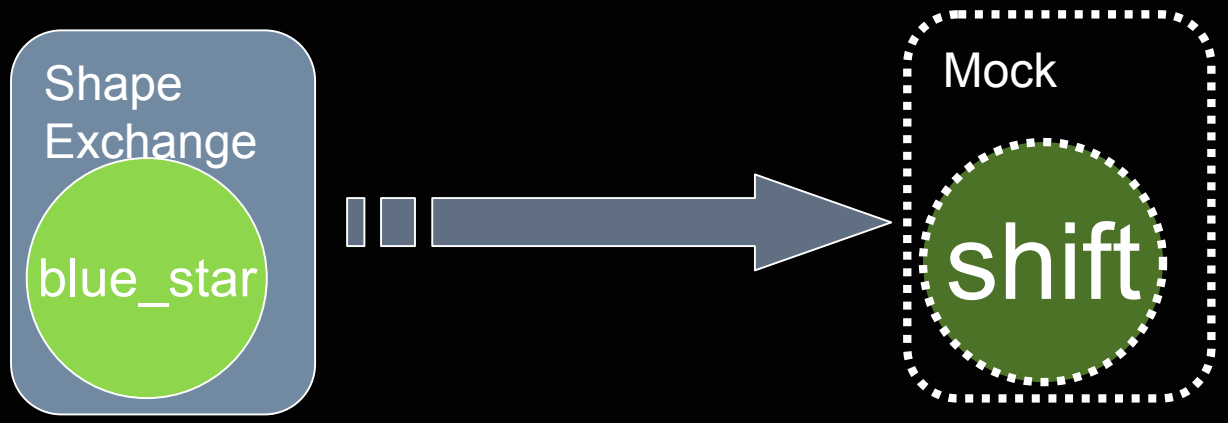
You get the idea

untitled 2

```
1  def test_makes_blue_star
2    # setup...
3    connect_to_db
4    load_shape_definitions_from_sap
5    create_shape_dependency_1
6    #....
7
8    assert_equal(:blue_star, ShapeExchange.blue_star)
9  end
10
```

Mock the interaction?

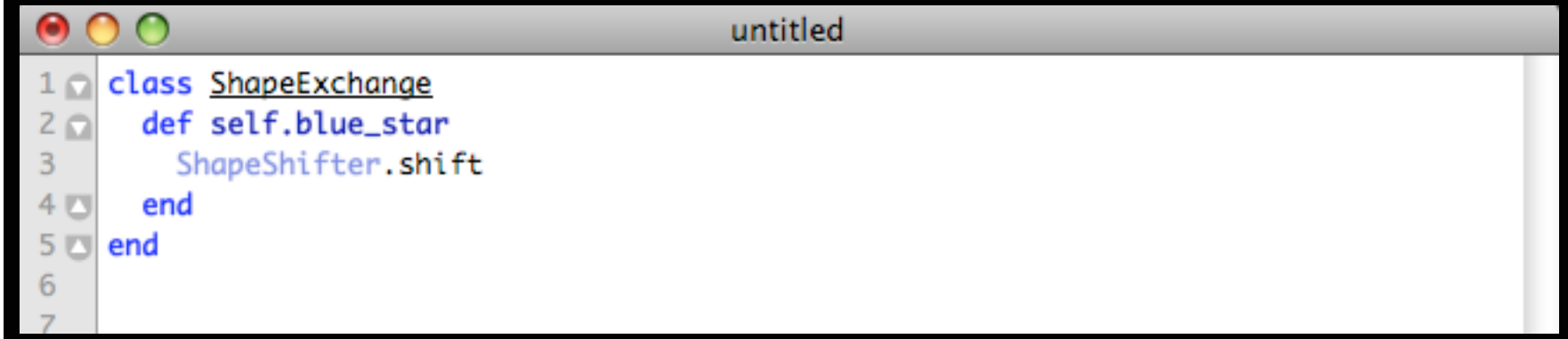
Red



untitled 2

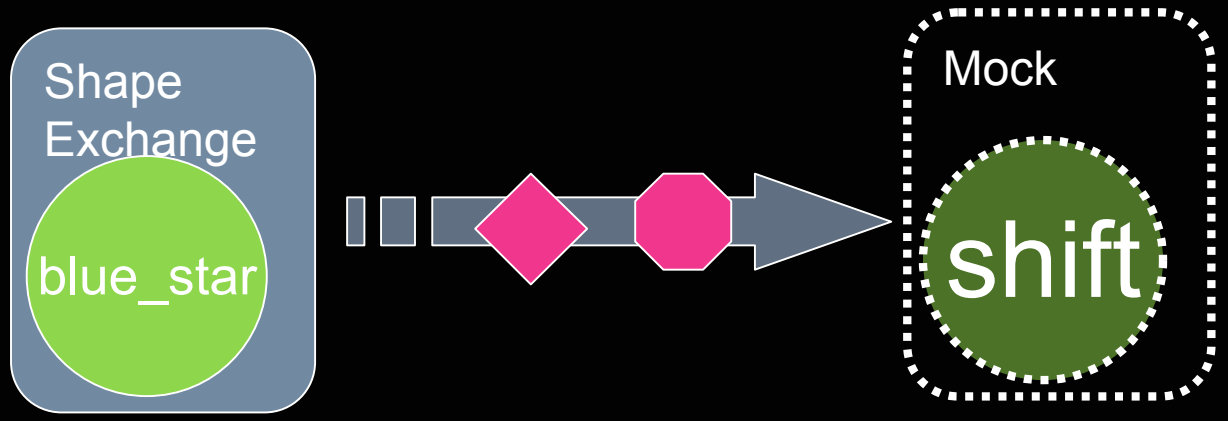
```
1 def test_makes_blue_star
2   ShapeShifter.expects(:shift)
3   ShapeExchange.blue_star
4 end
```

Green



Crash

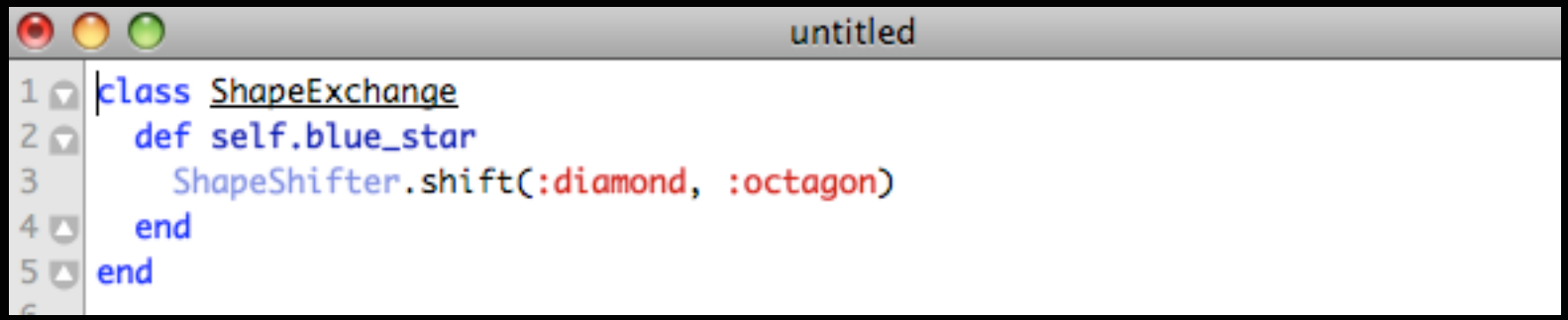
Red



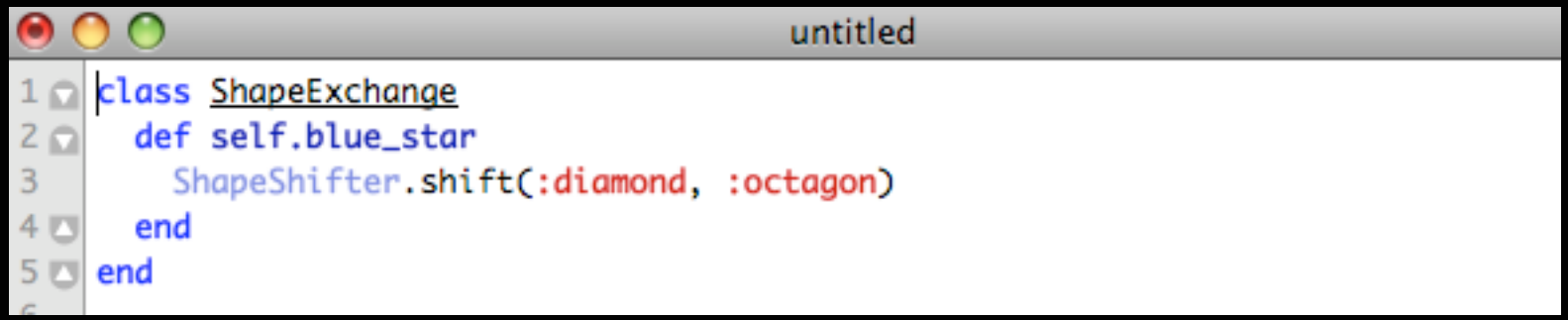
untitled 2

```
1 def test_makes_blue_star
2   ShapeShifter.expects(:shift).with(:diamond, :octagon)
3   ShapeExchange.blue_star
4 end
```

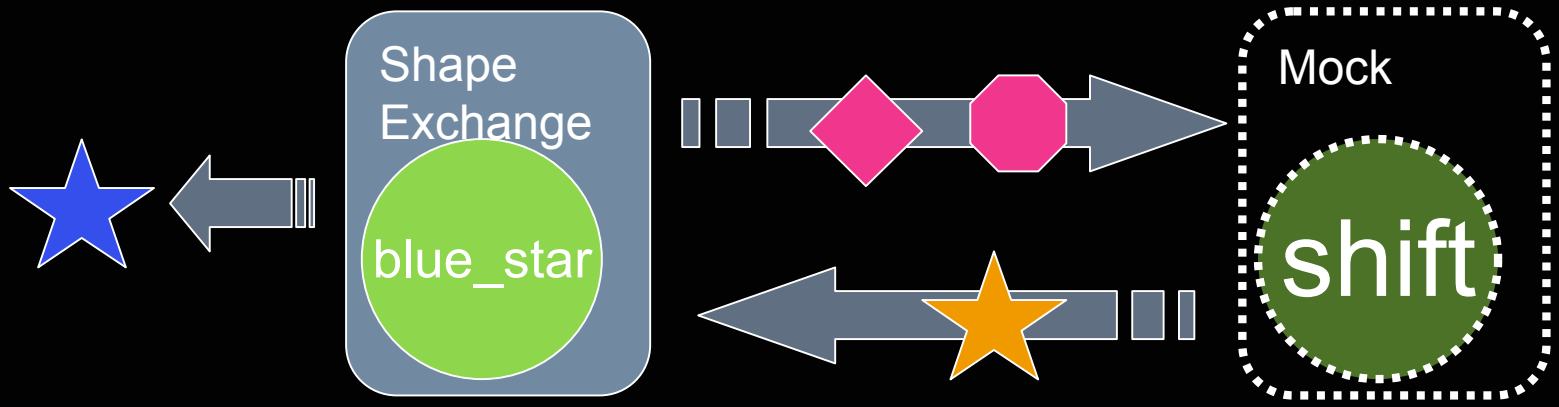
Green



Incomplete



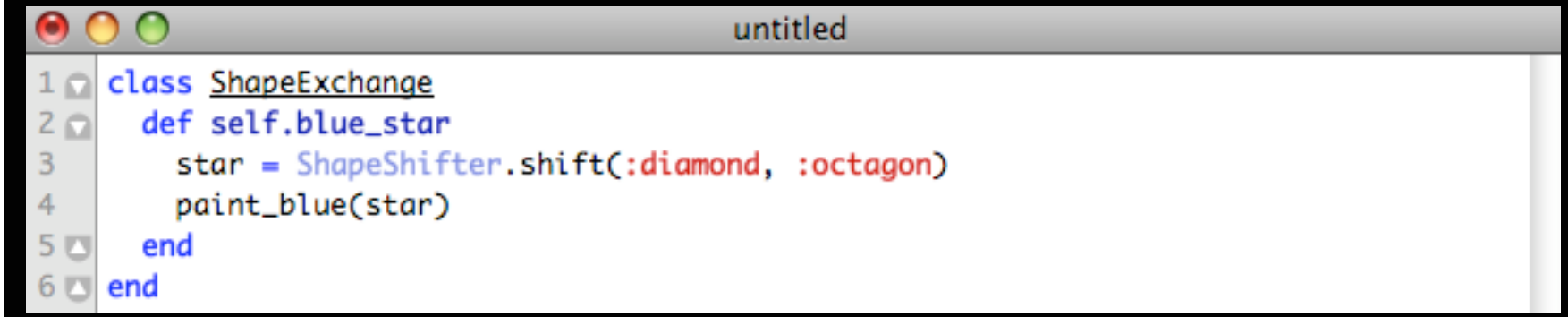
Red



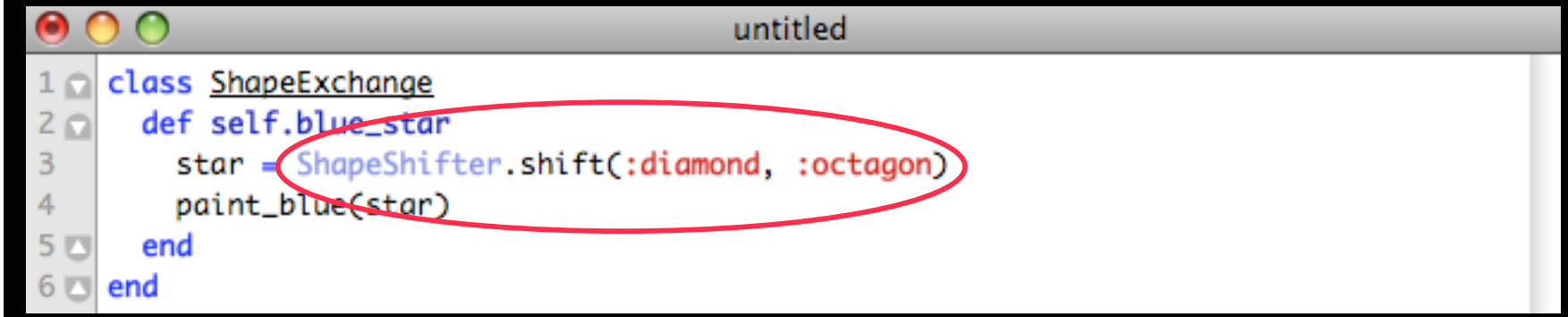
untitled 2

```
1 def test_makes_blue_star
2   ShapeShifter.expects(:shift).with(:diamond, :octagon).returns(:star)
3   assert_equal(:blue_star, ShapeExchange.blue_star)
4 end
```

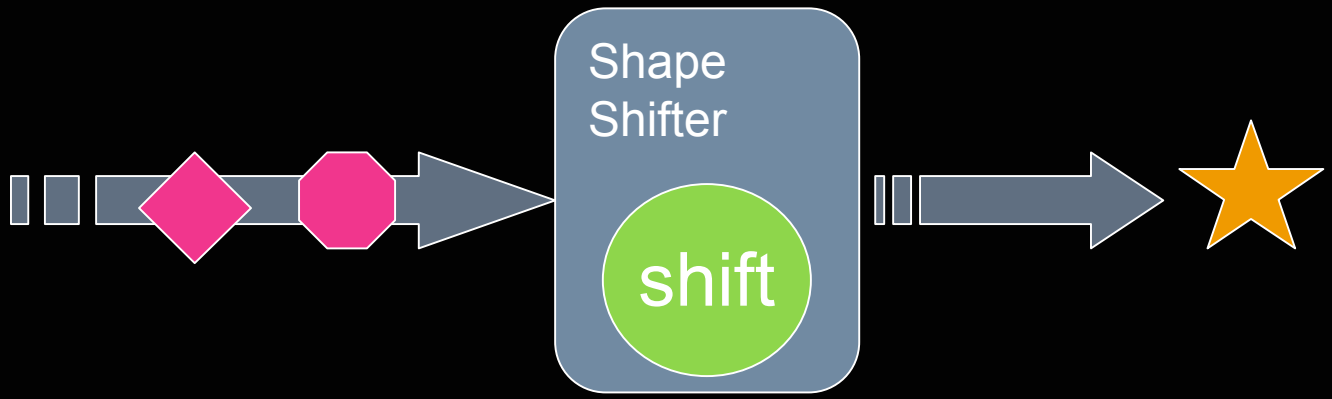
Green



Where?

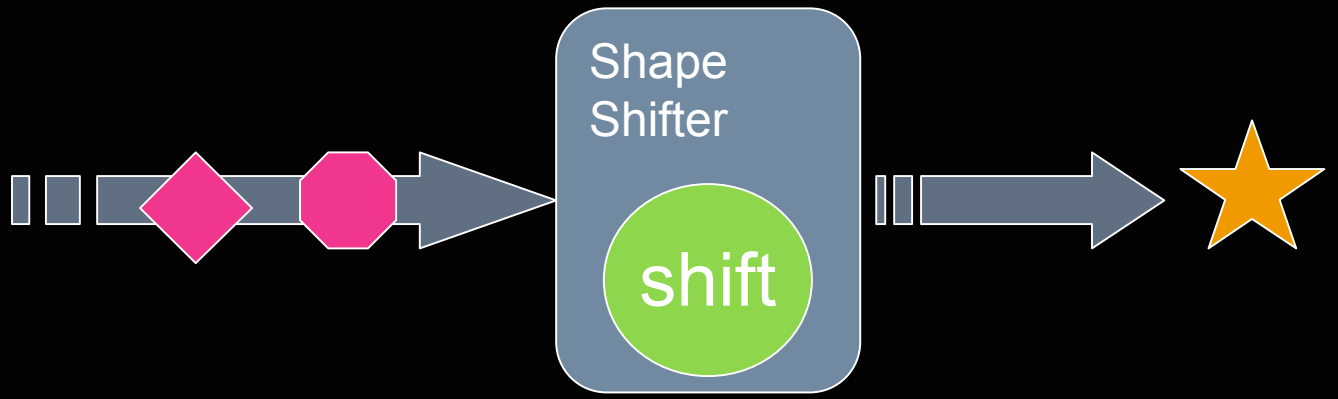


Can we prove this happens?



?

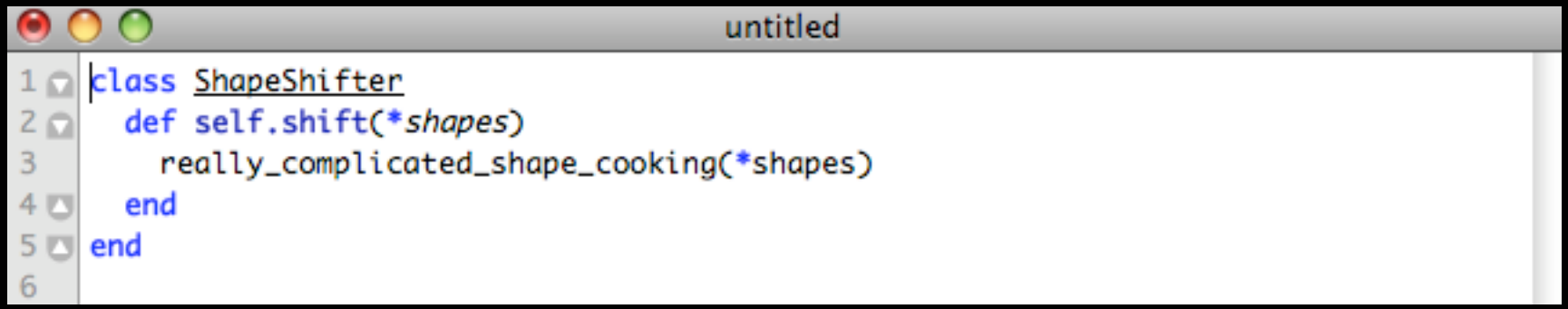
Red



untitled 2

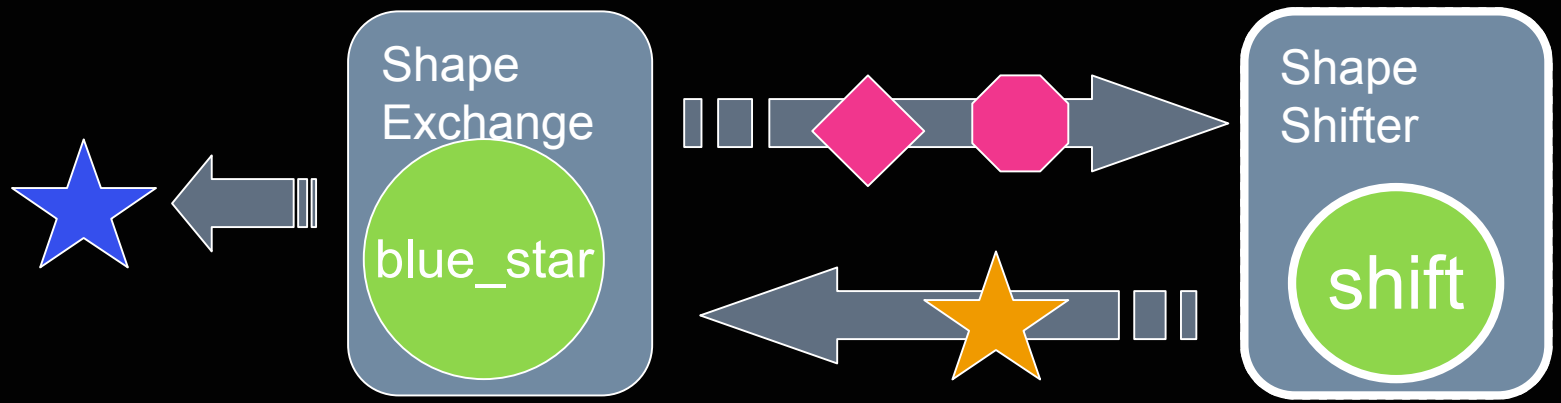
```
1  def test_shifts_start_from_diamond_and_octagon
2    assert_equal(:star, ShapeShifter.shift(:diamond, :octagon))
3  end
4
```

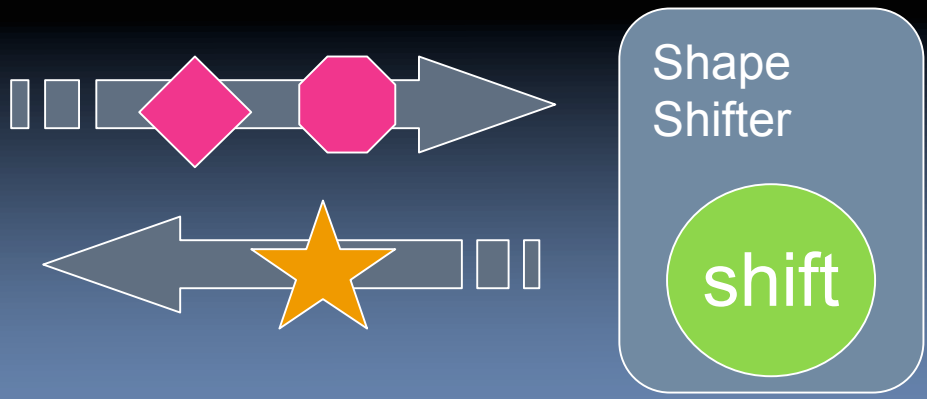
Green



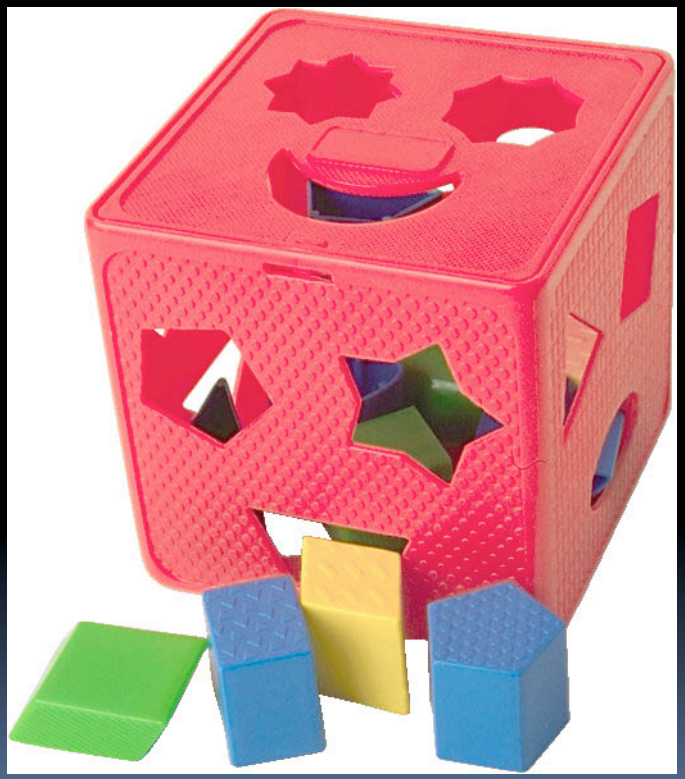
Reduced technical debt

Synthesizing a bigger test





Synthesis



Works well with

Existing tools

- RSpec
- Mocha
- zentest
- autotest

Testing Practices

- TDD
- BDD

Synthesis overview

Used through the Rake task Synthesis::Task

```
33 Synthesis::Task.new('synthesis:test:rspec') do |t|
34   t.adapter = :rspec
35   t.pattern = 'test_project/rspec/*_spec.rb'
36 end
```

Synthesis overview

First pass

- Collect mocked expectations

Second Pass

- Correlate mocked expectations with REAL calls to REAL objects.

```ruby
require File.dirname(__FILE__) + "/helper"

class DataBranderTest < Test::Unit::TestCase
  def test_saves_branded_with_storage
    storage = Storage.new('whatever')
    storage.expects(:save).with('Hello from Prague')
    data_brander = DataBrander.new(storage)
    data_brander.save_branded('Hello')
  end
end

```

```ruby
class DataBrander
  def initialize(storage)
    @storage = storage
  end

  def save_branded(data)
    @storage.save("#{data} from Prague")
  end
end
```

```ruby
require File.dirname(__FILE__) + "/helper"

class StorageTest < Test::Unit::TestCase
  def test_saves_to_file
    Storage.new('test-file.txt').save('hello')
    assert_equal('hello', File.read('test-file.txt'))
  ensure
    FileUtils.rm_f('test-file.txt') if File.file?('test-file.txt')
  end
end
```

```ruby
class Storage
  def initialize(filename)
    @filename = filename
  end

  def save(data)
    File.open(@filename, 'w') {|f| f << data}
  end
end
```

```
RakeMate v2.0.0 running on Ruby v1.8.6 (/usr/bin/ruby)
>>> /Users/gmalamid/devel/ruby/whatever_code/synthesis_examples/Rakefile

(in /Users/gmalamid/devel/ruby/whatever_code/synthesis_examples)
Loaded suite /Library/Ruby/Gems/1.8/gems/rake-0.8.1/lib/rake/rake_test_loader
Started
..
Finished in 0.002593 seconds.

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

```
1  require "simple"
2
3  storage = Storage.new("hello.txt")
4  data_brander = DataBrander.new(storage)
5  data_brander.save_branded("Hello")
```

```ruby
class Storage
  def initialize(filename)
    @filename = filename
  end

  def save(data)
    File.open(@filename, 'w') {|f| f << data}
  end
end
```

```ruby
class Storage
  def initialize(filename)
    @filename = filename
  end

  def save(data, mode)
    File.open(@filename, mode) {|f| f << data}
  end
end
```

```ruby
require File.dirname(__FILE__) + "/helper"

class StorageTest < Test::Unit::TestCase
  def test_saves_to_file
    Storage.new('test-file.txt').save('hello', 'w')
    assert_equal('hello', File.read('test-file.txt'))
  ensure
    FileUtils.rm_f('test-file.txt') if File.file?('test-file.txt')
  end
end
```

```
RakeMate v2.0.0 running on Ruby v1.8.6 (/usr/bin/ruby)
>>> /Users/gmalamid/devel/ruby/whatever_code/synthesis_examples/Rakefile

(in /Users/gmalamid/devel/ruby/whatever_code/synthesis_examples)
Loaded suite /Library/Ruby/Gems/1.8/gems/rake-0.8.1/lib/rake/rake_test_loader
Started
..
Finished in 0.001725 seconds.

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

```
1  require "simple"
2
3  storage = Storage.new("hello.txt")
4  data_brander = DataBrander.new(storage)
5  data_brander.save_branded("Hello")
```

ArgumentError: wrong number of arguments (1 for 2)

> method save in **data_brander.rb** at line 7
> method save_branded in **data_brander.rb** at line 7
> *at top level* in **main.rb** at line 5

Program exited.

```ruby
class DataBrander
  def initialize(storage)
    @storage = storage
  end

  def save_branded(data)
    @storage.save("#{data} from Prague")
  end
end
```

```
17      desc "Synthesize Simple"
18      Synthesis::Task.new('simple') do |t|
19        t.pattern = 'simple/test/*_test.rb'
20      end
```

```
[Synthesis] Collecting expectations...
Loaded suite /usr/bin/rake
Started
..
Finished in 0.004425 seconds.

2 tests, 2 assertions, 0 failures, 0 errors
[Synthesis] Verifying expectation invocations...
Loaded suite /usr/bin/rake
Started
..
Finished in 0.002309 seconds.

2 tests, 2 assertions, 0 failures, 0 errors
[Synthesis]
[Synthesis] Tested Expectations:
[Synthesis]
[Synthesis] Untested Expectations:
[Synthesis] Storage.new.save(String) in ./simple/test/data_brander_test.rb:6:in `test_saves_branded_with_storage'
[Synthesis]
[Synthesis] Ignoring:
[Synthesis]
[Synthesis] FAILED.
```

```
1  require File.dirname(__FILE__) + "/helper"
2
3  class DataBranderTest < Test::Unit::TestCase
4    def test_saves_branded_with_storage
5      storage = Storage.new('whatever')
6      storage.expects(:save).with('Hello from Prague', 'w')
7      data_brander = DataBrander.new(storage)
8      data_brander.save_branded('Hello')
9    end
10 end
11
```

```ruby
class DataBrander
  def initialize(storage)
    @storage = storage
  end

  def save_branded(data)
    @storage.save("#{data} from Prague", 'w')
  end
end
```

```
[Synthesis] Collecting expectations...
Loaded suite /usr/bin/rake
Started
..
Finished in 0.002872 seconds.

2 tests, 2 assertions, 0 failures, 0 errors
[Synthesis] Verifying expectation invocations...
Loaded suite /usr/bin/rake
Started
..
Finished in 0.001661 seconds.

2 tests, 2 assertions, 0 failures, 0 errors
[Synthesis]
[Synthesis] Verified 1 expectations
[Synthesis] SUCCESS.
```

```ruby
class CreatePeople < Sequel::Migration
  def up
    create_table :people do
      primary_key :id
      varchar :first_name
      varchar :last_name
    end
  end

  def down
    drop_table :people
  end
end

DB = Sequel.sqlite
CreatePeople.apply(DB, :up)
```

```ruby
require File.dirname(__FILE__) + "/helper"

class GreeterTest < Test::Unit::TestCase
  def test_says_hello
    person = stub(:first_name => 'John', :last_name => 'Doe')
    Person.expects(:find_by_first_name).with("John").returns(person)
    assert_equal("Hello, John Doe", Greeter.say_hello("John"))
  end
end
```

```ruby
class Greeter
  def self.say_hello(first_name)
    person = Person.find_by_first_name(first_name)
    "Hello, #{person.first_name} #{person.last_name}"
  end
end
```

```ruby
require File.dirname(__FILE__) + "/helper"

class PersonTest < Test::Unit::TestCase
  def test_find_by_name
    Person.create(:first_name => "John", :last_name => "Doe")
    assert_equal("Doe", Person.find_by_first_name("John").last_name)
  end
end
```

```
1 class Person < Sequel::Model
2 end
```

```
RakeMate v2.0.0 running on Ruby v1.8.6 (/usr/bin/ruby)
>>> /Users/gmalamid/devel/ruby/whatever_code/synthesis_examples/Rakefile

(in /Users/gmalamid/devel/ruby/whatever_code/synthesis_examples)
Loaded suite /Library/Ruby/Gems/1.8/gems/rake-0.8.1/lib/rake/rake_test_loader
Started
..
Finished in 0.002974 seconds.

2 tests, 3 assertions, 0 failures, 0 errors
```

```ruby
#!/usr/bin/env ruby
require File.dirname(__FILE__) + "/sql"
require File.dirname(__FILE__) + "/sql/database"

Person.create(:first_name => 'John', :last_name => 'Doe')

p Greeter.say_hello("John")
```

```
RubyMate r8136 running Ruby r1.8.6
(/System/Library/Frameworks/Ruby.framework/Versions/1.8/usr/bin
/ruby)
>>> main.rb

"Hello, John Doe"
```

```ruby
class CreatePeople < Sequel::Migration
  def up
    create_table :people do
      primary_key :id
      varchar :first_name
      varchar :last_name
    end
  end

  def down
    drop_table :people
  end
end

DB = Sequel.sqlite
CreatePeople.apply(DB, :up)
```

```ruby
class CreatePeople < Sequel::Migration
  def up
    create_table :people do
      primary_key :id
      varchar :given_name
      varchar :last_name
    end
  end

  def down
    drop_table :people
  end
end

DB = Sequel.sqlite
CreatePeople.apply(DB, :up)
```

```ruby
require File.dirname(__FILE__) + "/helper"

class PersonTest < Test::Unit::TestCase
  def test_find_by_name
    Person.create(:given_name => "John", :last_name => "Doe")
    assert_equal("Doe", Person.find_by_given_name("John").last_name)
  end
end
```

```
RakeMate v2.0.0 running on Ruby v1.8.6 (/usr/bin/ruby)
>>> /Users/gmalamid/devel/ruby/whatever_code/synthesis_examples/Rakefile

(in /Users/gmalamid/devel/ruby/whatever_code/synthesis_examples)
Loaded suite /Library/Ruby/Gems/1.8/gems/rake-0.8.1/lib/rake/rake_test_loader
Started
..
Finished in 0.003556 seconds.

2 tests, 3 assertions, 0 failures, 0 errors
```

Sequel::Error::InvalidStatement: INSERT INTO people (first_name, last_name) VALUES ('John', 'Doe') table people has no column named first_name

method execute_insert	in **sqlite.rb** at line 50
method insert	in **sqlite.rb** at line 171
method save!	in **record.rb** at line 238
method save	in **validations.rb** at line 12
method create	in **record.rb** at line 133
method transaction	in **sqlite.rb** at line 124
method transaction	in **database.rb** at line 594
method transaction	in **sqlite.rb** at line 124
method hold	in **connection_pool.rb** at line 61
method transaction	in **sqlite.rb** at line 118
method create	in **record.rb** at line 131
at top level	in **main.rb** at line 5

```
[Synthesis] Collecting expectations...
Loaded suite /usr/bin/rake
Started
..
Finished in 0.008957 seconds.

2 tests, 3 assertions, 0 failures, 0 errors
[Synthesis] Verifying expectation invocations...
Loaded suite /usr/bin/rake
Started
..
Finished in 0.00263 seconds.

2 tests, 3 assertions, 0 failures, 0 errors
[Synthesis]
[Synthesis] Tested Expectations:
[Synthesis]
[Synthesis] Untested Expectations:
[Synthesis] Person.find_by_first_name(String) in ./sql/test/greeter_test.rb:6:in `test_says_hello'
[Synthesis]
[Synthesis] Ignoring:
[Synthesis]
[Synthesis] FAILED.
```

```ruby
require File.dirname(__FILE__) + "/helper"

class GreeterTest < Test::Unit::TestCase
  def test_says_hello
    person = stub(:given_name => 'John', :last_name => 'Doe')
    Person.expects(:find_by_given_name).with("John").returns(person)
    assert_equal("Hello, John Doe", Greeter.say_hello("John"))
  end
end
```

```ruby
class Greeter
  def self.say_hello(given_name)
    person = Person.find_by_given_name(given_name)
    "Hello, #{person.given_name} #{person.last_name}"
  end
end
```

```
RakeMate v2.0.0 running on Ruby v1.8.6 (/usr/bin/ruby)
>>> /Users/gmalamid/devel/ruby/whatever_code/synthesis_examples/Rakefile

(in /Users/gmalamid/devel/ruby/whatever_code/synthesis_examples)
[Synthesis] Collecting expectations...
Loaded suite /usr/bin/rake
Started
..
Finished in 0.004433 seconds.

2 tests, 3 assertions, 0 failures, 0 errors
[Synthesis] Verifying expectation invocations...
Loaded suite /usr/bin/rake
Started
..
Finished in 0.00482 seconds.

2 tests, 3 assertions, 0 failures, 0 errors
[Synthesis]
[Synthesis] Verified 1 expectations
[Synthesis] SUCCESS.
```

```
RubyMate r8136 running Ruby r1.8.6
(/System/Library/Frameworks/Ruby.framework/Versions/1.8/usr/bin
/ruby)
>>> main.rb

"Hello, John Doe"
```

Synthesis

- http://synthesis.rubyforge.org/

- stuart.caborn@thoughtworks.com
- george.malamidis@thoughtworks.com