Agile Development with Ruby

Premshree Pillai

YAHOO!

ppillai@yahoo-inc.com

What is Agile?

- Test-driven development
- Incremental development
- Continual delivery
- Software over documentation

How does Ruby facilitate agile development?

- Test-driven development (using unit testing modules)
  - User stories
  - Limited number of stories considered at a time
  - Unit tests written before actual code
  - Tests for the specification
- More testing
  - Introspection

If you're new to Ruby...

- Interpreted
- Pure OO
- Dynamic, strong typed
- Often compared to Perl and Python

Testing: Looking at objects

```
class MyClass
  …
end

obj = MyClass.new

ObjectSpace.each_object(MyClass) { |x|
  p x }

Returns <MyClass:0x812e658>
```

Testing: Looking inside objects

```
class  MyClass
  def foo
      …
  end
  def bar
      …
  end
end

Obj = MyClass.new
p obj.methods
```

Returns ["foo", "bar"]

Testing: Looking inside objects (contd.)

- `obj.respond_to? (method_to_test)`
- `obj.id`
- `obj.class`
- `obj.kind_of? type_to_test`
- `obj.instance_of? class_to_test`

Testing: Classes and methods

- Classes
  - `MyClass.private_instance_methods`
  - `MyClass.protected_instance_methods`
  - `MyClass.public_instance_methods`
  - `MyClass.singleton_methods`
  - `MyClass.constants`
  - `MyClass.superclass.constants`
- Methods
  - `MyClass_obj.send("method_name")`
  - `MyClass_obj.method("method_name")`

Unit testing with Ruby: Modules

- Test::Unit

```
require 'test/unit'
class TC_ClassToTest < Test::Unit::TestCase
    def test_method_to_test
          assert_equal("expected output",
                                "actual_output",
                                "some_string")
    end
end
```

- RubyUnit

Mock objects

```
mock = Object.new
mock.extend(Test::Unit::Assert)
def mock.method_called_by_testee
  assert_equal("expected output",
               "actual output",
               "some_string")
end

testObject.something_with(mock)
```

RubyUnit

- Initial setup:

```
require 'runit/testcase'
require 'runit/cui/testrunner'
require 'myclass' # class under test (CUT)
```

- Test class (TC):

```
class Testing_class < RUNIT::TestCase
    …
end
```

RubyUnit (contd.)

- Methods of test class begin with `test_`

```
class Testing_class <
RUNIT::TestCase
   def test_1
         …
   end
end
```

RubyUnit (contd.)

- RUNIT::TestCase methods
  - `assert_fail(message)`
  - `assert(boolean, message="")`
  - `assert_equal(expected, actual, message="")`
  - `assert_same(expected, actual, message="")`
  - `assert_nil(obj, message="")`
  - `assert_respond_to(method, obj, message="")`
  - `assert_kind_of(c, obj, message="")`

RubyUnit (contd.)

- RUNIT::TestCase methods (contd.)
  - `assert_instance_of(c, obj , message="")`
  - `assert_match(str, re, message="")`
  - `assert_send(obj1, op, *args)`
  - …

RubyUnit: Running tests

- Running a specific test

```
RUNIT::CUI::TestRunner.run(Testing_c
  lass.new("some_test_method"))
```

- Running a test suite

```
RUNIT::CUI::TestRunner.run(Testi
ng_class.suite)
```

RubyUnit: Sample template test code

```
if __FILE__ == $0
   require 'runit/testcase'
   require 'runit/cui/testrunner'
   require 'runit/testsuite'

   class Testing_class < RUNIT::TestCase
       def test_method1
             mine = MyClass.new
             …
       end
   end

   RUNIT::CUI::TestRunner.run(Testing_class.suite)
end
```

Building an app using RubyUnit

```
require 'runit/testcase'
require 'runit/cui/testrunner'
require 'runit/testsuite'

class TC_NewLineToHtmlBreak < RUNIT::TestCase
  def test_NewLineToHtmlBreak
      nl2br = NewLineToHtmlBreak.new
      assert_equal("bombay\nrocks!", nl2br.convert,
  "newlines should've been converted to <br>s")
  end
  RUNIT::CUI::TestRunnr.run(TC_NewLineToHtmlBreak.suit
  e)
end
```

Building an app using RubyUnit (contd.)

```
class NewLineToHtmlBreak
  def convert(str)
     return str.gsub!("\n", "<br
  />")
  end
end
```

Building an app using RubyUnit (contd.)

```
require 'runit/testcase'
require 'runit/cui/testrunner'
require 'runit/testsuite'
require 'NewLineToHtmlBreak'

class TC_NewLineToHtmlBreak < RUNIT::TestCase
  def test_NewLineToHtmlBreak
      nl2br = NewLineToHtmlBreak.new
      assert_equal("bombay\nrocks!", nl2br.convert,
  "newlines should've been converted to <br>s")
  end
  RUNIT::CUI::TestRunnr.run(TC_NewLineToHtmlBreak.suit
  e)
end
```

Unit testing web apps

- MVC architecture
  - Decoupling models and contollers – mock objects
  - Automated feeds to controllers (from views)
  - Method interception

References

- Ruby language: http://ruby-lang.org/
- Test::Unit: http://testunit.talbott.ws/
- Ruby on Rails: http://www.rubyonrails.com/
- I write about Ruby when I'm not drunk: http://www.livejournal.com/~premshree

Thengyoo!

Questions?