**Red Team**

**Stargazer**

**Automatic Telescope Control System**

**Testing Plan**

**Team Members:**

Jason Dunscombe

Rob Grmek

Robert Smith

**Instructor:**

**Youry Khmelevsky**

**Course:**

**COSC 470**

**Date:**

**November 8th, 2009**

# ****Revision History****

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 10/21/2009 | 1.0 | Document created. | Robert, Rob |
| 11/08/2009 | 1.1 | Added unit testing information. | Rob |

# ****Table of Contents****

**Page**

|  |  |
| --- | --- |
| Unit Tests | 1 |
| Acceptance Tests | 3 |

# Unit Tests

We will be using the extreme programming style for all development in this project. This means we will be creating unit tests, using Ruby’s built-in unit testing, before writing any actual code. These tests will build on each other to form our test suite.

For a tutorial on using Ruby’s unit testing with Ruby on Rails, see here:

* <http://www.tutorialspoint.com/ruby-on-rails-2.1/rails-unit-testing.htm>
* <http://guides.rubyonrails.org/testing.html>
* <http://andrzejonsoftware.blogspot.com/2007/05/15-tdd-steps-to-create-rails.html>

**Example #1: creating a test case for a model (from SPIKE project):**

In ~/Blog/test/unit/post\_test.rb:

|  |
| --- |
| require 'test\_helper'  class PostTest < ActiveSupport::TestCase  def test\_should\_not\_save\_post\_without\_title  post = Post.new  assert !post.save, "Saved the post without a title"  end  end |

To run that single test method:

|  |
| --- |
| # from project directory root  cd test  ruby unit/post\_test.rb -n test\_should\_not\_save\_post\_without\_title |

To run the single test case:

|  |
| --- |
| # from project directory root  cd test  ruby unit/post\_test.rb |

To run all unit tests and functional tests:

|  |
| --- |
| rake test |

To run only all of the unit tests:

|  |
| --- |
| rake test:units |

To run only all of the functional tests:

|  |
| --- |
| rake test:functionals |

To run only recent tests for models and controllers that have been modified in last 10 minutes:

|  |
| --- |
| rake test:recent |

**Example #2: creating a test case for a controller (from SPIKE project):**

In ~/Blog/test/functional/post\_controller\_test.rb:

|  |
| --- |
| require 'test\_helper'  class PostsControllerTest < ActionController::TestCase  test "should get index" do  get :index  assert\_response :success  assert\_not\_nil assigns(:posts)  end  test "should get new" do  get :new  assert\_response :success  end  test "should create post" do  assert\_difference('Post.count') do  post :create, :post => { :name => "Bob", :title => "Something"}  end  assert\_redirected\_to post\_path(assigns(:post))  assert\_equal "Your post is saved!", flash[:notice]  end  test "should show post" do  get :show, :id => posts(:one).to\_param  assert\_response :success  end  test "should get edit" do  get :edit, :id => posts(:one).to\_param  assert\_response :success  end  test "should update post" do  put :update, :id => posts(:one).to\_param, :post => { }  assert\_redirected\_to post\_path(assigns(:post))  end  test "should destroy post" do  assert\_difference('Post.count', -1) do  delete :destroy, :id => posts(:one).to\_param  end  assert\_redirected\_to posts\_path  end  end |

# Acceptance Tests

We are still trying to get in contact with the customer, but once we do, we will work with him to create a set of acceptance tests that will confirm that our software meets his needs.