**Red Team**

**Stargazer**

**Automatic Telescope Control System**

**Project Plan**

**Team Members:**

Rob Grmek

Robert Smith

**Instructor:**

**Youry Khmelevsky**

**Course:**

**COSC 471**

**Date:**

**March 10th, 2010**

# ****Revision History****

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 10/16/2009 | 1.0 | Document created. | Robert, Rob, Jason |
| 11/1/2009 | 1.1 | Added information for 2nd iteration. | Rob |
| 11/8/2009 | 1.2 | Added additional information for release #1 and #2. | Rob |
| 11/13/2009 | 1.3 | Minor updates. | Rob |
| 11/16/2009 | 1.4 | Added metrics for release #1 | Rob |
| 11/24/2009 | 1.5 | Added metrics for release #2 | Rob, Robert |
| 11/29/2009 | 1.6 | Added some changes. | Rob |
| 1/7/2010 | 1.7 | Plans for the second semester. | Rob |
| 1/20/2010 | 1.8 | Updated plans. | Rob |
| 1/22/2010 | 1.9 | Updated plans. | Rob |
| 2/8/2010 | 2.0 | Updated plans. | Robert |
| 2/11/2010 | 2.1 | Small update. | Rob |
| 2/24/2010 | 2.2 | Update for release #5. | Rob |
| 3/8/2010 | 2.3 | Updated plans and risk management plan for digital camera issue. | Rob |
| 3/10/2010 | 2.4 | Added analysis for later iterations. | Rob |

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

**Page**

|  |  |
| --- | --- |
| Introduction | 1 |
| Risk Management Plan | 2 |
| Iteration 1: Inception, Partial Elaboration and Initial Design | 3 |
| Iteration 2: SPIKE Project | 5 |
| Iteration 3: Authentication and Authorization | 6 |
| Iteration 4: Image Gallery, Presentation, and Mobile Devices | 8 |
| Iteration 5: Automated Scheduling and Telescope Positioning | 10 |
| Iteration 6: Libraries and GUI Enhancements | 12 |
| Iteration 7: Mobile Views | 15 |
| Iteration 8: Image Capturing | 17 |
| Iteration 9: The Final Release | 19 |

# ****Project Plan****

This document covers the planning process for the design, development and deliverance of our Stargazer projectfor our customers Alexander and Nan.

Note:

Durations in our planning process is measured in units. 1 unit is equal to 15 minutes.

Task durations tracked with Time Tracker: <http://timetracker.wrconsulting.com>

Furthermore, the analysis of possible risks with corresponding risk management solutions which attempt to either minimize or avoid the risk from becoming an issue is included as well in this document.

Also, further project management can be done using the following tool: <http://project.youry.net>

Additional metrics for the web application can be dynamically attained through the following website: <http://getcaliper.com/> using the application’s Git repository (git://github.com/RedTeamCOSC470/Stargazer.git).

We retrieved metrics for the web application mainly using Roodi (<http://roodi.rubyforge.org/>) and Flog (<http://ruby.sadi.st/Flog.html>).

# Risk Management Plan

|  |  |  |  |
| --- | --- | --- | --- |
| Risk | Probability | Effect | Solution |
| Underestimate the time the project will take. | Medium to high. | Serious – we run out of time. | Remove non-essential features. |
| Illness to team members. | Low to medium. | Catastrophic – We are low on manpower as it is. | Assign and distribute their tasks to the other team member. |
| Not understanding requirements. | Medium to high. | Serious -The system behaves incorrectly. | Keep in close contact with the customer. |
| Scheduling conflicts between team members. | Low to medium. | Serious – Individual programming is not XP programming. | Schedule well before hand. |
| No available lab space. | Medium. | Tolerable – need to find other areas to work. | Use laptops in meeting rooms or other areas. |
| Server or technical issues. | Low to medium. | Catastrophic – May lose work. | Backup regularly; use a versioning control system. |
| Campus network problems. | High. | Serious – Reduced productivity, difficulty accessing development server. | Work directly on development server or work locally on a laptop computer. |
| Lack of time at end of semester. | High. | Tolerable – Reduced ability to work outside of class/lab time. | Use class/lab time as productively as possible. |
| Client unavailable or unresponsive. | Medium. | Catastrophic – No customer input. | Use alternative methods of communication (phone or email).  May have to select a different project. |
| Not receiving equipment necessary to complete project. | Medium. | Catastrophic – Not able to complete project to requirements. | Use alternative devices (such as a different digital camera) and try to use a generalized solution that should work for both devices. |

Probabilities can be: low, moderate, high.

Effects can be: tolerable, serious and catastrophic.

# Iteration 1: Inception, Partial Elaboration and Initial Design

Start date: October 21st, 2009

End date: October 28th, 2009

We have selected our project and have begun drafting initial documentation related to the project and set up a development environment.

Also, we planned to run through several Ruby on Rails related tutorials to learn how to use the development, how to use versioning control with Git and how to use unit testing with Ruby’s built-in unit testing.

### Task Duration and Dependencies:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task** | **#** | **Estimated Duration** | **Dependency** | **Actual Duration** |
| Create initial vision | 1 | 2 | - | 2 |
| Create project plan document | 2 | 2 | - | 3 |
| Add business case and product info to vision | 3 | 4 | 1 | 3 |
| Add architecture to vision | 4 | 1 | 1 | 1 |
| Add risks to project plan | 5 | 2 | 2 | 1 |
| Research image recognition technologies | 6 | 24 | - | - |
| Research hardware I/O specifications | 7 | 24 | - | - |
| Research development environment components | 8 | 24 | - | 12 |
| Set up development environment | 9 | 16 | 8 | 8 (inc.) |
| Create tutorial plan | 10 | 2 | - | 2 |
| Create initial developer's guide | 11 | 8 | - | 6 |
| Create use case diagram | 12 | 2 | - | 2 |
| Write fully dressed use cases | 13 | 6 | 12 | 4 |
| Complete tutorials using our framework | 14 | 32 | 8, 9 | - |
| Draft acceptance test | 15 | 4 | - | - |
| Add installation guide information | 16 | 4 | 11 | 4 |

### Tutorial Plan:

Setting up a development environment on Windows:

<http://www.akitaonrails.com/2009/1/13/the-best-environment-for-rails-on-windows>

Setting up a development environment on Fedora 11:

<http://www.technetra.com/2009/04/22/howto-setting-up-ruby-on-rails-for-fedora-10-and-11/>

Rails Guides:

1. <http://guides.rubyonrails.org>
2. <http://www.tutorialspoint.com/ruby-on-rails-2.1/index.htm>
3. <http://www.meshplex.org/wiki/Ruby/Ruby_on_Rails_programming_tutorials>

Rails Development Tutorials:

1. <http://guides.rubyonrails.org/getting_started.html>
2. <http://www.rails4days.pwp.blueyonder.co.uk/Rails4Days.pdf>
3. <http://www.tutorialized.com/view/tutorial/Image-uploads-and-resizing-for-Rails-models-with-mini-magick/19070>

Git Tutorial:

<http://harryseldon.thinkosphere.com/2009/01/14/git-and-rails-a-detailed-tutorial-including-plugins-submodules-development-and-production>

# Iteration 2: SPIKE Project

Start date: October 28th, 2009

End date: November 6th, 2009

We have researched the technologies related to the project in iteration 1 and can now begin a short one week SPIKE project to learn the Ruby on Rails framework. The SPIKE project we created was that of a blogging site.

The SPIKE project must:

* Have login system and authentication; since this is important to learn how to do for our real project.
* Be committed regularly to our GitHub repository and use source control management (SCM).
* Have unit testing; we’ll need to use test-driven development (TDD) in our real project, so it is key to learn the built-in ruby testing tool.

### List of Tasks:

|  |  |  |
| --- | --- | --- |
| **Task** | **Estimated Duration** | **Actual Duration** |
| SPIKE: Unit testing tutorials | 4 | 3 |
| SPIKE: Use Git to push/pull project to/from GitHub | 2 | 2 |
| SPIKE: Post scaffolding tutorial | 6 | 5 |
| SPIKE: Comment and tag functionality tutorial | 6 | 6 |
| SPIKE: Collect SPIKE code for submission | 2 | 3 |
| Design: Create domain model | 4 | 4 |
| Design: Create architecture diagram | 2 | 2 |
| Documentation: Add using Git information to developer's guide | 4 | 3 |
| Documentation: Update project plan for release #1 and #2 | 4 | 6 |
| Documentation: Add architecture diagram to developer's guide | 2 | 2 |
| Documentation: Get extra user stories from customer | 3 | 4 |
| Documentation: Prioritize the user stories | 1 | 1 |
| Documentation: Update installation information | 4 | 4 |
| Documentation: Update vision | 2 | 2 |
| Set up development environment | 8 | 6 |
| Test telescope interface/drivers | 12 | 26 |
| Test telescope programs (Autostar, ASCOM) | 8 | 10 |
| Total: |  | 89 |

### Milestones:

* SPIKE completion: Nov. 6, 2009

# Iteration 3: Authentication and Authorization

Start date: November 6th, 2009

End date: November 18th, 2009

From our knowledge gained from the SPIKE project, we can begin development of the web application portion of our project.

In this one week iteration, we plan to implement the following functionality:

* Authentication; force the user to login in order to access the system.
* Scheduling telescope position; show a form to allow the user to input a time, date, coordinates, etc which contains all the information necessary to properly position the telescope.
* Review scheduling logs; show entries for all schedules that have been created in the past, present and future.
* Online help documentation; gives users information on using the system.

### List of Tasks:

|  |  |  |
| --- | --- | --- |
| **Task** | **Estimated Duration** | **Actual Duration** |
| Documentation: Update testing plan | 3 | 2 |
| Create web system: Add authentication | 12 | 11 |
| Create web system: Add scheduling form (with validation/tests) | 16 | 18 |
| Create web system: Create scheduling review logs | 2 | 1 |
| Create web system: Add online help documentation | 12 | 4 |
| Create web system: Add authorization | 16 | 23 |
| Documentation: Create change requests document | 1 | 1 |
| Documentation: Update all other documents. | 24 | 18 |
| Design: Create class diagram | 2 | 8 |
| Design: Update use case diagram | 1 | 1 |
| Refactoring | 4 | 4 |
| Total: | 93 | 89 |

### Milestones:

* Release #1: Nov. 18, 2009

#### Metrics:

Unit Testing:

* Unit tests: 2
* Test cases: 18
* Assertions: 31

Implementation:

* Views: 15 (including partials)
* Models: 3
* Controllers: 4

# Iteration 4: Image Gallery, Presentation and Mobile Devices

Start date: November 18th, 2009

End date: December 4th, 2009

In this iteration, we are to primarily build onto the web systems functionality. We plan to create:

* Stylesheets; create separate CSS for both mobile device users and PC users for easier visibility and usability.
* An image gallery; show latest composite images from the telescope.
* HTTPS/HTTP for admin/user
* Set up production server (use Apache as production web server)
* Changes to schedule input forms and model

Also, extensive refactoring must be done in this iteration.

### List of Tasks:

|  |  |  |
| --- | --- | --- |
| **Task** | **Estimated Duration** | **Actual Duration** |
| Web system: Add separate stylesheets for PC users and mobile device users | 8 | 8 |
| Web system: Add gallery section - display latest images from telescope | 16 | 22 |
| Web system: HTTPS/HTTP for admin/user | 8 | 6 |
| Web system: Set up production server | 8 | 10 |
| Update all documentation and create design diagrams | 40 | 16 |
| Web system: Refactoring | 20 | 6 |
| Web system: Change schedule input forms | 12 | 16 |
| Web system: Fix unit tests | 4 | 4 |
| Web system: Fix CSS issues | 4 | 5 |
| Create PowerPoint Presentation | 6 | 10 |
| Total: | 100 | 103 |

### Milestones:

* Release #2: Nov. 25, 2009
* Release #3 and Final Presentation: Dec. 4, 2009

#### Metrics:

Unit Testing:

* Unit tests: 3
* Test cases: 17
* Assertions: 29

Implementation:

* Views: 26 (including partials)
* Models: 4
* Controllers: 6

# Iteration 5: Automated Scheduling and Telescope Positioning

Start date: January 6th, 2010

End date: February 4th, 2010

Web system functions to be implemented and tasks to be completed:

* Job scheduling; telescope positioning schedules created through the web system should correspond to a job schedule on the OS to run at the specific time as entered by the user.
* Some refactoring left from last iteration needs to be completed.
* Current online help documentation needs to be added.

Tasks centered on the telescope controlling application:

* Allow ability to park the telescope.
* Allow telescope movement to specific celestial coordinates.

### List of Tasks:

|  |  |  |
| --- | --- | --- |
| **Task** | **Estimated Duration** | **Actual Duration** |
| Documentation: Documentation updates | 20 | 45 |
| Documentation: Metrics gathering | 6 | 8 |
| Documentation: Draft user acceptance test | 4 | 0 (inc.) |
| Web system: Add new online help documentation to website | 12 | 12 |
| Telescope-controlling application: Include telescope parking ability | 4 | 3 |
| Telescope-controlling application: Include telescope movement ability from entered celestial coordinates | 8 | 28 |
| Web system: Code refactoring | 6 | 7 |
| Web system: Research Rails plug-ins to handle scheduling | 8 | 8 |
| Database: Create triggers/batch file to run application and additional actions | 8 | 8 (inc.) |
| Database: Download and install Oracle 10g XE on telescope server | 6 | 6 |
| Database: Install Oracle client on web server | 4 | 4 |
| Web system: Configure Rails environment to now use the Oracle 10g XE database | 4 | 3 |
| Database: Test connections and scheduling jobs | 8 | 6 |
| Telescope-controlling application: Create unit tests | 10 | 0 (inc.) |
| Total: | 108 | 138 |

### Milestones:

* Release #4: Feb. 4th, 2010

#### Web Application Metrics:

Unit Testing:

* Unit tests: 3
* Test cases: 16
* Assertions: 28

Implementation:

* Views: 31 (including partials)
* Models: 4
* Controllers: 6

Design problems - 1

Problems:LOC - 1:374

Size Measurements:

|  |  |  |
| --- | --- | --- |
| Lines of Application Code | Lines of Unit Test Code | Code to Test Ratio |
| 374 | 238 | 1:0.6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Lines | LOC | Classes | Methods | Methods per class | LOC per method |
| Libraries | 9 | 8 | 0 | 0 | 0 | 0 |
| App/controllers | 464 | 282 | 6 | 34 | 5 | 6 |
| App/models | 148 | 53 | 4 | 5 | 1 | 8 |
| App/helpers | 39 | 31 | 0 | 5 | 0 | 4 |
| Test/unit | 260 | 141 | 8 | 19 | 2 | 5 |
| Test/functional | 114 | 90 | 5 | 0 | 0 | 0 |
| Test/performance | 9 | 7 | 1 | 1 | 1 | 5 |
| Total | 1043 | 612 | 24 | 64 | 2 | 7 |

Complexity Measurements (using Flog):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| File | Total score | Methods | Average score | Highest score |
| [/app/controllers/images\_controller.rb](http://getcaliper.com/caliper/tool?tool=flog&repo=git://github.com/RedTeamCOSC470/Stargazer.git#_app_controllers_images_controller_rb) | 124 | 8 | 16 | 36 |
| [/app/controllers/schedules\_controller.rb](http://getcaliper.com/caliper/tool?tool=flog&repo=git://github.com/RedTeamCOSC470/Stargazer.git#_app_controllers_schedules_controller_rb) | 151 | 8 | 19 | 35 |
| [/app/models/schedule.rb](http://getcaliper.com/caliper/tool?tool=flog&repo=git://github.com/RedTeamCOSC470/Stargazer.git#_app_models_schedule_rb) | 60 | 5 | 12 | 32 |
| [/app/controllers/users\_controller.rb](http://getcaliper.com/caliper/tool?tool=flog&repo=git://github.com/RedTeamCOSC470/Stargazer.git#_app_controllers_users_controller_rb) | 65 | 8 | 8 | 16 |
| [/app/helpers/schedules\_helper.rb](http://getcaliper.com/caliper/tool?tool=flog&repo=git://github.com/RedTeamCOSC470/Stargazer.git#_app_helpers_schedules_helper_rb) | 14 | 1 | 14 | 14 |
| [/app/controllers/user\_sessions\_controller.rb](http://getcaliper.com/caliper/tool?tool=flog&repo=git://github.com/RedTeamCOSC470/Stargazer.git#_app_controllers_user_sessions_controller_rb) | 20 | 3 | 7 | 12 |
| [/app/controllers/application\_controller.rb](http://getcaliper.com/caliper/tool?tool=flog&repo=git://github.com/RedTeamCOSC470/Stargazer.git#_app_controllers_application_controller_rb) | 47 | 9 | 5 | 8 |
| [/app/helpers/layout\_helper.rb](http://getcaliper.com/caliper/tool?tool=flog&repo=git://github.com/RedTeamCOSC470/Stargazer.git#_app_helpers_layout_helper_rb) | 9 | 3 | 3 | 4 |
| [/app/models/user.rb](http://getcaliper.com/caliper/tool?tool=flog&repo=git://github.com/RedTeamCOSC470/Stargazer.git#_app_models_user_rb) | 2 | 1 | 2 | 2 |
| [/app/models/image.rb](http://getcaliper.com/caliper/tool?tool=flog&repo=git://github.com/RedTeamCOSC470/Stargazer.git#_app_models_image_rb) | 2 | 1 | 2 | 2 |
| [/lib/stargazer/version.rb](http://getcaliper.com/caliper/tool?tool=flog&repo=git://github.com/RedTeamCOSC470/Stargazer.git#_lib_stargazer_version_rb) | 2 | 1 | 2 | 2 |
| [/app/controllers/info\_controller.rb](http://getcaliper.com/caliper/tool?tool=flog&repo=git://github.com/RedTeamCOSC470/Stargazer.git#_app_controllers_info_controller_rb) | 1 | 1 | 1 | 1 |

Total score for all methods: 497.1

Average score for all methods: 10.1

# Iteration 6: Libraries and GUI Enhancements

Start date: February 4th, 2010

End date: February 25th, 2010

Note: “Reading Break” from February 15th to 19th.

Web system functions to be implemented and tasks to be completed:

* Integrate libraries which contain celestial objects of interest (e.g. certain planets) so as to allow users an alternative to entering coordinates manually.
* Pagination added to tables such as the schedule log.
* Increase usability by adding search filtering and JavaScript calendars, decomposing the online help by separate topics and adding AJAX auto complete text fields for the celestial object inputs.
* Validations need to ensure that no two schedules can occur at the same time.
* Allow users to park the telescope via the web system.

Tasks centered on the telescope controlling application:

* Allow for the use of the celestial objects to be entered from the library.
* Unit tests need to be completed.

Modeling that needs to be updated for the web application:

* Class diagrams.

### List of Tasks:

|  |  |  |
| --- | --- | --- |
| **Task** | **Estimated Duration** | **Actual Duration** |
| Documentation: Documentation updates | 16 | 16 |
| Integrate libraries (which have celestial objects of interest) to web application | 20 | 24 |
| Web system: Add scheduled job for parking telescope | 20 | 18 |
| Web system: Add pagination to website tables | 10 | 6 |
| Web system: Bug fixing and refactoring | 14 | 12 |
| Web system: Schedule duration validations | 10 | 16 |
| Web system: Add search features | 7 | 6 |
| Web system: Add JavaScript calendars | 6 | 6 |
| Web system: Online help enhancements | 2 | 3 |
| Web system: Integrate libraries (which have celestial objects of interest) with web site | 20 | 26 |
| Telescope-controlling application: Add celestial object inputs into the application | 4 | 8 |
| Documentation: Create class diagram | 4 | 8 |
| Total: | 133 | 149 |

### Milestones:

* Release #5: Feb. 25th, 2010
* Complete telescope parking: Feb. 25th, 2010
* Complete hardware interface: Feb. 11th, 2010

#### Web Application Metrics:

Unit Testing:

* Unit tests: 3
* Test cases: 16
* Assertions: 28

Implementation:

* Views: 38 (including partials)
* Models: 5
* Controllers: 8

Design problems - 1

Problems:LOC - 1:430

Size Measurements:

|  |  |  |
| --- | --- | --- |
| Lines of Application Code | Lines of Unit Test Code | Code to Test Ratio |
| 430 | 262 | 1:0.6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Lines | LOC | Classes | Methods | Methods per class | LOC per method |
| Libraries | 9 | 8 | 0 | 0 | 0 | 0 |
| app/helpers | 43 | 35 | 0 | 5 | 0 | 5 |
| app/models | 227 | 87 | 5 | 7 | 1 | 10 |
| app/controllers | 533 | 300 | 8 | 41 | 5 | 5 |
| test/functional | 130 | 102 | 7 | 0 | 0 | 0 |
| test/unit | 292 | 153 | 11 | 19 | 1 | 6 |
| test/performance | 9 | 7 | 1 | 1 | 1 | 5 |

Complexity Measurements (using Flog): 9.5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| File | Total score | Methods | Average score | Highest score |
| [/app/models/schedule.rb](http://getcaliper.com/caliper/tool?tool=flog&repo=git://github.com/RedTeamCOSC470/Stargazer.git#_app_models_schedule_rb) | 108 | 8 | 14 | 64 |
| [/app/controllers/images\_controller.rb](http://getcaliper.com/caliper/tool?tool=flog&repo=git://github.com/RedTeamCOSC470/Stargazer.git#_app_controllers_images_controller_rb) | 124 | 8 | 16 | 36 |
| [/app/controllers/schedules\_controller.rb](http://getcaliper.com/caliper/tool?tool=flog&repo=git://github.com/RedTeamCOSC470/Stargazer.git#_app_controllers_schedules_controller_rb) | 118 | 9 | 13 | 26 |
| [/app/controllers/users\_controller.rb](http://getcaliper.com/caliper/tool?tool=flog&repo=git://github.com/RedTeamCOSC470/Stargazer.git#_app_controllers_users_controller_rb) | 71 | 8 | 9 | 16 |
| [/app/helpers/schedules\_helper.rb](http://getcaliper.com/caliper/tool?tool=flog&repo=git://github.com/RedTeamCOSC470/Stargazer.git#_app_helpers_schedules_helper_rb) | 14 | 1 | 14 | 14 |
| [/app/controllers/user\_sessions\_controller.rb](http://getcaliper.com/caliper/tool?tool=flog&repo=git://github.com/RedTeamCOSC470/Stargazer.git#_app_controllers_user_sessions_controller_rb) | 20 | 3 | 7 | 12 |
| [/app/controllers/application\_controller.rb](http://getcaliper.com/caliper/tool?tool=flog&repo=git://github.com/RedTeamCOSC470/Stargazer.git#_app_controllers_application_controller_rb) | 47 | 9 | 5 | 8 |
| [/app/controllers/celestial\_objects\_controller.rb](http://getcaliper.com/caliper/tool?tool=flog&repo=git://github.com/RedTeamCOSC470/Stargazer.git#_app_controllers_celestial_objects_controller_rb) | 7 | 1 | 7 | 7 |
| [/app/helpers/layout\_helper.rb](http://getcaliper.com/caliper/tool?tool=flog&repo=git://github.com/RedTeamCOSC470/Stargazer.git#_app_helpers_layout_helper_rb) | 9 | 3 | 3 | 4 |
| [/app/controllers/info\_controller.rb](http://getcaliper.com/caliper/tool?tool=flog&repo=git://github.com/RedTeamCOSC470/Stargazer.git#_app_controllers_info_controller_rb) | 2 | 1 | 2 | 2 |
| [/app/models/image.rb](http://getcaliper.com/caliper/tool?tool=flog&repo=git://github.com/RedTeamCOSC470/Stargazer.git#_app_models_image_rb) | 2 | 1 | 2 | 2 |
| [/app/models/user.rb](http://getcaliper.com/caliper/tool?tool=flog&repo=git://github.com/RedTeamCOSC470/Stargazer.git#_app_models_user_rb) | 2 | 1 | 2 | 2 |
| [/app/models/celestial\_object.rb](http://getcaliper.com/caliper/tool?tool=flog&repo=git://github.com/RedTeamCOSC470/Stargazer.git#_app_models_celestial_object_rb) | 2 | 1 | 2 | 2 |
| [/lib/stargazer/version.rb](http://getcaliper.com/caliper/tool?tool=flog&repo=git://github.com/RedTeamCOSC470/Stargazer.git#_lib_stargazer_version_rb) | 2 | 1 | 2 | 2 |
| [/app/controllers/telescope\_controller.rb](http://getcaliper.com/caliper/tool?tool=flog&repo=git://github.com/RedTeamCOSC470/Stargazer.git#_app_controllers_telescope_controller_rb) | 1 | 1 | 1 | 1 |

Total score for all methods: 530.0

Average score for all methods: 9.5

#### Analysis:

The plan for this iteration was to mainly implement the image capturing functionality as this is a key feature and a main requirement of our system. Since we have received delays in receiving the necessary equipment (such as the camera), this had to be postponed further.

Furthermore, the following tasks which were planned to be done this iteration had to be rescheduled to the next iteration as we had run out of time:

* Create unit tests for telescope-controlling application
* Create sequence diagrams
* Create system sequence diagrams
* Create activity diagrams
* Create state diagrams

# Iteration 7: Mobile Views

Start date: February 25th, 2010

End date: March 11th, 2010

Web system functions to be implemented and tasks to be completed:

* Increase functionality, utility and usability for mobile device users.
* Add validations for celestial object/coordinate inputs; they need to be mutually exclusive and jointly exhaustive.
* Online help; add administration section and refactor design.

Also, a significant amount of modeling and documentation needs to be done.

### List of Tasks:

|  |  |  |
| --- | --- | --- |
| **Task** | **Estimated Duration** | **Actual Duration** |
| Documentation: Documentation updates | 24 |  |
| Web system: Add validations for celestial object/coordinates | 7 |  |
| Web system: Add mobile device views and additional functionality | 16 |  |
| Web system: Fix bug with testing environment | 10 |  |
| Web system: Add administration section to online help | 3 | 5 |
| Web system: Refactor online help | 5 |  |
| Documentation: Draft acceptance test | 4 |  |
| Documentation: Create system sequence diagrams | 8 |  |
| Documentation: Create activity diagrams | 10 |  |
| Documentation: Create state diagrams | 10 |  |
| Documentation: Create sequence diagrams | 8 |  |
| Total: | 105 |  |

### Milestones:

* Release #7: Mar. 11th, 2010

#### Web Application Metrics:

Unit Testing:

* Unit tests: 3
* Test cases: 16
* Assertions: 28

Implementation:

* Views: 38 (including partials)
* Models: 5
* Controllers: 8

#### Analysis:

Originally, this iteration was planned to deal with the issue of image capturing. This had to be postponed until next iteration as we had delays with receiving the equipment necessary to start this task (such as acquiring a digital camera).

Also, some implementation needed to be carried over from last iteration that was left undone. An example is the validations dealing with celestial objects and to finish integrating the celestial library functionality (as the Schedules table’s trigger needed to be adjusted and a new batch file created).

Furthermore, some relatively trivial tasks such as refactoring and updating of online help and documentation proved to take up more time than anticipated. It wasn’t a result of technical issues but because of communication issues. Time was also spent fruitlessly trying to rectify errors with the web application’s test environment which won’t be correctly generated from the schema file which is generated from the migration files. This error is still to be resolved but only affects the running of unit tests.

# Iteration 8: Image Capturing

Start date: ?

End date: ?

Web system functions to be implemented and tasks to be completed:

* Add image copying from the telescope server to the web server.

Tasks centered on the telescope controlling application:

* Image capturing: the application now is able to control the camera and able to take images.
* Image capturing: solution must be generalized and must work on both the Canon 30D we received from the college and also the Sony A900 that the customer uses.
* Image processing: once the many images have been taken, a composite image is assembled.

### List of Tasks:

|  |  |  |
| --- | --- | --- |
| **Task** | **Estimated Duration** | **Actual Duration** |
| Documentation: Documentation updates | 16 |  |
| Web system: Form improvements (on mobile devices) | 8 |  |
| Web system: Create rake task for image copying | 25 |  |
| Web system: Integrate camera values to camera calls | 6 |  |
| Web system: Fix bug with testing environment | 10 |  |
| Telescope-controlling application: Create program to handle image taking with specific parameters | 18 |  |
| Telescope-controlling application: Integrate camera | 2 |  |
| Telescope-controlling application: Implement functionality for compiling composite images | 18 |  |
| Telescope-controlling application: Create unit tests | 10 |  |
| Total: | 113 |  |

### Milestones:

* Release #7: Mar. ?th, 2010

### Analysis:

This iteration was originally intended to deal with implementing a level of image recognition. New objects would be found by processing images captured and based on referring those images with previous images or known objects in the library. However, this has to be abandoned due to equipment delays and time constraints. There was also a possibility of implementing manual positioning of the telescope, but this was outside the scope of the project and no time is available to pursue this feature. Consequently, no notifications need to be sent to users as there will be no image recognition which means no email server and programs need to be added.

Now, after receiving a digital camera, the implementation of image capturing must be realized. While we do not have access to the customer’s Sony A900 DSLR digital camera, we have access to a less powerful digital camera of a different make and model (Canon 30D).

# Iteration 9: The Final Release

Start date: ?

End date: April 8th, 2010

Extensive refactoring must be done in this iteration and the final product needs to be deployed. As well, system has to be fully tested to ensure functional correctness and user satisfaction.

Furthermore, all project-related metrics and statistics need to be gathered and documentation needs to be finalized.

### List of Tasks:

|  |  |  |
| --- | --- | --- |
| **Task** | **Estimated Duration** | **Actual Duration** |
| Documentation: Finalize documentation | 24 |  |
| Documentation: Compile final metrics and project statistics | 4 |  |
| Deployment: Deploy systems to production environment | 4 |  |
| Refactoring: Final refactoring | 20 |  |
| Testing: System testing | 12 |  |
| Testing: Integration testing | 8 |  |
| Testing: Fix functional testing | 12 |  |
| Testing: Final user acceptance test | 4 |  |
| Presentation: Create PowerPoint presentation | 10 |  |
| Total: | 98 |  |

### Milestones:

* Final Release: Apr. 8th, 2010
* Internal Presentation: Apr. 8th, 2010
* Corrective Coding Ceases: Apr. 20th, 2010
* Final Presentation: Apr. 26th, 2010