SPAZIO

FINAL REPORT

Team Members:
Mark Viola
Mike Petrella
Hyeong Ook Yu
ZiXian Chen
Michael Liu

Table of Contents

1. Product Backlog

1.1 Plugin Support

Priority: High
Cost: 12 Story Points

As Hanno (an administrator), I want plugin API support for the application, where I can create, add, modify plugins written in a programming language that the application can use to extract data from other catalogs.

Tasks:

1. Build/find a class library to make it easier to deal with web requests.
2. Create plugin interface.
3. Create a XML parser that will take a dictionary and output a XML format text, with fields in the same order as dictionary.
4. Create serializer class and add plugin loading functionality
5. Create two default plugins for exoplanet.eu and NASA Exoplanet Archive

1.2 Standardizing Units

Priority: High
Cost: 8 Story Points

As Hanno (an administrator), I want each plugin to be able to standardize units of measurements found in its catalogue to one that I specify in the settings of the application.

Tasks:

1. Find/build a unit conversion library
   - Create a way of making general conversion (i.e., specify original unit and convert to whatever is defined in the settings)

1.3 Scheduling a Check for Updates

Priority: High
Cost: 5 Story Points

As Hanno (an administrator), I want the application to check the other exoplanet databases it is tracking, for new updates, on a specified interval (default is daily at 11:59pm).

Tasks:

1. Create a server framework that will run the plugins with threads
2. Have server constantly check system time

1.4 User Permissions

Priority: High
Cost: 5 Story Points

As Hanno (an administrator), I want to use GitHub repo collaborators as a way of distinguishing which user has access to use the application (manage pull-requests, run the server).

Tasks:

1. Create GitHub API management class, and add oauth2 login functionality
2. Add ability to check if user is a collaborator of the repo

---

1.5 Application Making Pull Requests

Priority: High
Cost: 20 Story Points

As Hanno (an administrator), I want the program to make pull requests to the OEC database when it finds changes in the databases that the application is monitoring, using an automated GitHub user account.

Tasks:

1. Add repo management functionality to GitHub API class
2. Add the ability to make pull requests to GitHub API class
3. Integrate GitHub API into a server so that pull-request are made with each new change
4. Add XML file reading to serializer
5. Add the ability to update a XML file with changes to serializer

---

1.6 Accepting/Rejecting Pending Pull Requests

Priority: High
Cost: 5 Story Points

As Irving (a user), I want to be able to accept, reject, or amend pull requests made by the application through a user interface by pressing a button on a pop-up window.

Tasks:

1. Add the ability to accept, reject, and change pull requests
2. Add pull-request to GUI
    - Add viewing
    - Add accepting/denying
    - Add sorting to display
    - Add editing

1.7 Merging Data from Plugins

Priority: High
Cost: 12 Story Points

As Hanno (an administrator), I want the data retrieved by a plugin to be merged with existing data in the Open Exoplanet Catalogue.

Tasks:

1. Write a class that merges two data types together
2. Turn the merged data back into an XML file

1.8 Source of the Update

Priority: Medium
Cost: 2 Story Points

As Hanno (an administrator), I want the website source of the new changes to be a comment in the pull request in the database.

Tasks:

1. Create a field that shows all of the publications related to a specific system. Shown when clicking on the system from the list of altered systems.
2. Make it mandatory for plugins to extract publication data

1.9 Edit Pending Pull Requests

Priority: Medium
Cost: 4 Story Points

As Irving (a user), I want to be able to edit a pending pull request through a user interface by pressing an edit button when shown a list of new changes.

Tasks:

1. Create an edit button that opens an interface that allows the user to edit a specified XML file.
2. Update the user’s current commit to the one that was edited using the interface.

1.10 Force a Check for Updates

Priority: Medium
Cost: 3 Story Points

As Hanno (an administrator), I want to be able to force the application to run even if it is not scheduled to check the databases.

Tasks:

1. Create a button in the Home screen that allows the application to check the tracked databases for updates.

1.11 Pull Request UI

Priority: Low
Cost: 14 Story Points

As Irving (a user), I want to be able to interact with the application using a user interface.

Tasks:

1. Use Glade to create the layout of the pull request UI window
2. Add backend implementation of the pull request UI using GTK

1.12 Plugin Management UI

Priority: Low
Cost: 14 Story Points

As Irving (a user), I want to be able to interact with the application using a user interface.

Tasks:

1. Use Glade to create the layout of the plugin management UI window
2. Add backend implementation of the plugin management UI using GTK

1.13 Bot Management UI

Priority: Low
Cost: 14 Story Points

As Irving (a user), I want to be able to interact with the application using a user interface.

Tasks:

1. Use Glade to create the layout of the bot management UI window
2. Add backend implementation of the bot management UI using GTK

1.14 Difference Between Updates

Priority: Low
Cost: 25 Story Points

As Irving (a user), I want to be able to click on a newly updated system from a list in the UI and have lines highlighted, showcasing which parts in the XML file have changed.

Tasks:

1. Add repo checking to GUI
    - Add the ability to see recent changes compared to the master branch
        - Create an interactable list of all recent changes
        - Create a GUI that will display changes of the XML files

---

1.15 Settings Form

Priority: Low
Cost: 15 Story Points

As Hanno (an administrator), I want to be able to change time controls, bot account settings, and conversion units, in a settings form in the application.

Tasks:

1. Use C# settings designer to create a settings form and maintain user settings
2. Add timed update controls to user settings
3. Add GitHub bot user account specification to user settings
4. Add a list of possible conversion units to user settings
5. Integrate user settings into the server with on demand changing
6. Add user settings to GUI

1.16 Changes from Deliverable 4

Due to the large scope that the “Create a User Interface” user story covered, it has been divided into the four user stories 1.13, 1.14, 1.15, and 1.17. Each of these stories defines a specific feature that the client wants when using a user interface. Additionally, after meeting with the client, we have been told that we no longer need to have a log of previous pull requests or a history of the planet/system edits, so these user stories have been removed from the product backlog.

2. Release Plan

Sprint 1:

- As Hanno (an administrator), I want plugin API support for the application, where I can create, add, modify plugins written in a programing language that the application can use to extract data from other catalogs.
- As Hanno (an administrator), I want the application to check other exoplanet databases for new changes on a specified interval (default is daily at 11:59pm).

Estimate: 19 Story Points

Sprint 2:

- As Hanno (an administrator), I want each to be able to standardize units of measurements found in its catalogue to one that I specify in the settings of the application. (**Finish remaining**)
- As Hanno (an administrator), I want to use GitHub repo collaborators as a way of distinguishing which user has access to use the application (manage pull-requests, run the server).
- As Hanno (an administrator), I want the program to make pull requests to the OEC database when it finds changes in the databases that the application is monitoring, using an automated GitHub user account.
- As Hanno (an administrator), I want the website source of the new changes to be a comment in the pull-request in the database.

Estimate: 30 Story Points

Sprint 3:

- As Irving (a user), I want to be able to interact with the application using a user interface. (**Half**)
- As Irving (a user), I want to be able to click on a newly updated system from a list in the UI and have lines highlighted, showcasing which parts in the XML file have changed.

Estimate: 25 Story Points

Sprint 4:

- As Irving (a user), I want to see the history of pull requests, on a user interface
- As Irving (a user), I want to be able to click on a newly updated system from a list in the UI and have lines highlighted, showcasing which parts in the XML file have changed.

Estimate: 25 Story Points

Sprint 5:

- As Hanno (an administrator), I want to the data retrieved by a plugin to be merged with existing data in the Open Exoplanet Catalogue.
- As Hanno (an administrator), I want to be able to force the application to run even if it is not scheduled to check the databases.
- As Irving (a user), I want to see the history of pull requests, on a user interface
- As Irving (a user), I want to see the planets that have been most recently added/edited by the application in the OEC, using the user interface.
- As Irving (a user), I want to be able to accept, reject, or amend pull requests made by the application through a user interface by pressing a button on a pop-up window.

Estimate: 31 Story Points

Sprint 6:

- As Irving (a user), I want to be able to edit a pending pull request through a user interface by pressing an edit button when shown a list of new changes.
- As Hanno (an administrator), I want to be able to change time controls, bot account settings, conversion units in a settings form in the application.
- Finish leftovers and testing

Estimate: 20 Story Points

3. Sprint Plan

3.1 Sprint Backlog (Sprint #5)

For the fifth sprint (11/14/2016 – 11/20/2016). This sprint will have a sprint velocity of 51 story points and will be 4 days long, starting on the Monday of this week. The following, details the user stories being completed and their corresponding tasks.

3.1.1 Merging Data from Plugins

Priority: High
Cost: 15 Story Points

As Hanno (an administrator), I want the data retrieved by a plugin to be merged with existing data in the Open Exoplanet Catalogue.

Task 1. Write a class that merges two data types together **(10 story points)**
Completed. Turn the merged data back into an XML file (5 story points)

3.1.2 Accepting/Rejecting Pending Pull Requests

Priority: High
Cost: 13 Story Points

As Irving (a user), I want to be able to accept, reject, or amend pull requests made by the application through a user interface by pressing a button on a pop-up window.

Task 2. Add the ability to accept, reject, and change pull requests **(13 story points)**

3.1.3 Pull Request UI

Priority: Low
Cost: 14 Story Points

As Irving (a user), I want to be able to interact with the application using a user interface to accept and reject pull requests.

Task 3. Create a UI window for pull requests using Use Glade to create the layout and GTK for the backend. **(14 story points)**

3.1.4 Plugin Management UI

Priority: Low
Cost: 5 Story Points

As Irving (a user), I want to be able to interact with the application using a user interface to manage the plugins that are being monitored.

Task 4. Create a UI window for plugin management using Use Glade to create the layout and GTK for the backend. **(5 story points)**

3.1.5 Bot Management UI

Priority: Low
Cost: 5 Story Points

As Irving (a user), I want to be able to interact with the application using a user interface to manage the settings and use of the GitHub bot.

Task 5. Create a UI window for the GitHub bot management using Use Glade to create the layout and GTK for the backend. **(5 story points)**

3.1.6 See History

Priority: Low
Cost: 4 Story Points

As Hanno (an administrator), I want to see the history of pull requests, on a user interface.

Task 6. Create a History screen that displays a list of the 20 most recent pull requests made by the GitHub bot. **(4 story points)**

Detailed Plan

	Nov 14	Nov 15	Nov 16	Nov 17	Nov 18	Nov 19	Nov 20
Task 1	5	5					
Task 2		4	4	5			
Task 3					6		3
Task 4					3	2	
Task 5					5		
Task 6							4

Actual

	Nov 14	Nov 15	Nov 16	Nov 17	Nov 18	Nov 19	Nov 20
Task 1		8					
Task 2			5	5		3	
Task 3					7		4
Task 4							
Task 5							
~~Task 6~~							

Legend:

Mark Viola
Mike Petrella
Hyeong Ook Yu
ZiXian Chen
Michael Liu

*Note: User story corresponding with Task 6 was removed after confirming with client that we no longer needed to implement the feature.

3.2 Sprint Backlog (Sprint #6)

For the sixth sprint (11/21/2016 – 11/27/2016). This sprint will have a sprint velocity of 39 story points and will be 4 days long, starting on the Monday of this week. The following, details the user stories being completed and their corresponding tasks.

3.2.1 Pull Request UI

Priority: Low
Cost: 20 Story Points

As Irving (a user), I want to be able to interact with the application using a user interface to accept and reject pull requests.

Task 1. Create a UI window for pull requests using Use Glade to create the layout and GTK for the backend. **(11 story points left)**

3.2.2 Plugin Management UI

Priority: Low
Cost: 5 Story Points

As Irving (a user), I want to be able to interact with the application using a user interface to manage the plugins that are being monitored.

Task 2. Create a UI window for plugin management using Use Glade to create the layout and GTK for the backend. **(5 story points)**

3.2.3 Force a Check for Updates

Priority: Medium
Cost: 3 Story Points

As Hanno (an administrator), I want to be able to force the application to run even if it is not scheduled to check the databases.

Task 3. Create a button in the Home screen that allows the application to check the tracked databases for updates. **(3 story points)**

3.2.4 Bot Management UI

Priority: Low
Cost: 5 Story Points

As Irving (a user), I want to be able to interact with the application using a user interface to manage the settings and use of the GitHub bot.

Task 4. Create a UI window for the GitHub bot management using Use Glade to create the layout and GTK for the backend. **(5 story points)**

3.2.5 Settings Form

Priority: Low
Cost: 15 Story Points

As Hanno (an administrator), I want to be able to change time controls, bot account settings, and conversion units, in a settings form in the application.

Task 5. Add user settings to GUI. **(15 story points)**

Detailed Plan

	Nov 21	Nov 22	Nov 23	Nov 24	Nov 25	Nov 26	Nov 27
Task 1		7	4				
Task 2		5					
Task 3			3				
Task 4				5			
Task 5				5	5		5

Actual

	Nov 21	Nov 22	Nov 23	Nov 24	Nov 25	Nov 26	Nov 27
Task 1	2	7	3				
Task 2							
Task 3		3					
Task 4			5				
Task 5							4

Legend:

Mark Viola
Mike Petrella
Hyeong Ook Yu
ZiXian Chen
Michael Liu

3.3 Sprint Backlog (Sprint #7)

For the seventh sprint (11/28/2016 – 12/1/2016). This sprint will have a sprint velocity of 20 story points and will be 4 days long, starting on the Monday of this week. The following, details the user stories being completed and their corresponding tasks.

3.3.1 Plugin Management UI

Priority: Low
Cost: 5 Story Points

As Irving (a user), I want to be able to interact with the application using a user interface to manage the plugins that are being monitored.

Task 1. Create a UI window for plugin management using Use Glade to create the layout and GTK for the backend

3.3.2 Settings Form

Priority: Low
Cost: 15 Story Points

As Hanno (an administrator), I want to be able to change time controls, bot account settings, and conversion units, in a settings form in the application.

Task 2. Add user settings to GUI

Detailed Plan

	Nov 28	Nov 29	Nov 30	Dec 1
Task 1		5		
Task 2		5	4	

Actual

	Nov 28	Nov 29	Nov 30	Dec 1
Task 1		4	5	
Task 2			7	

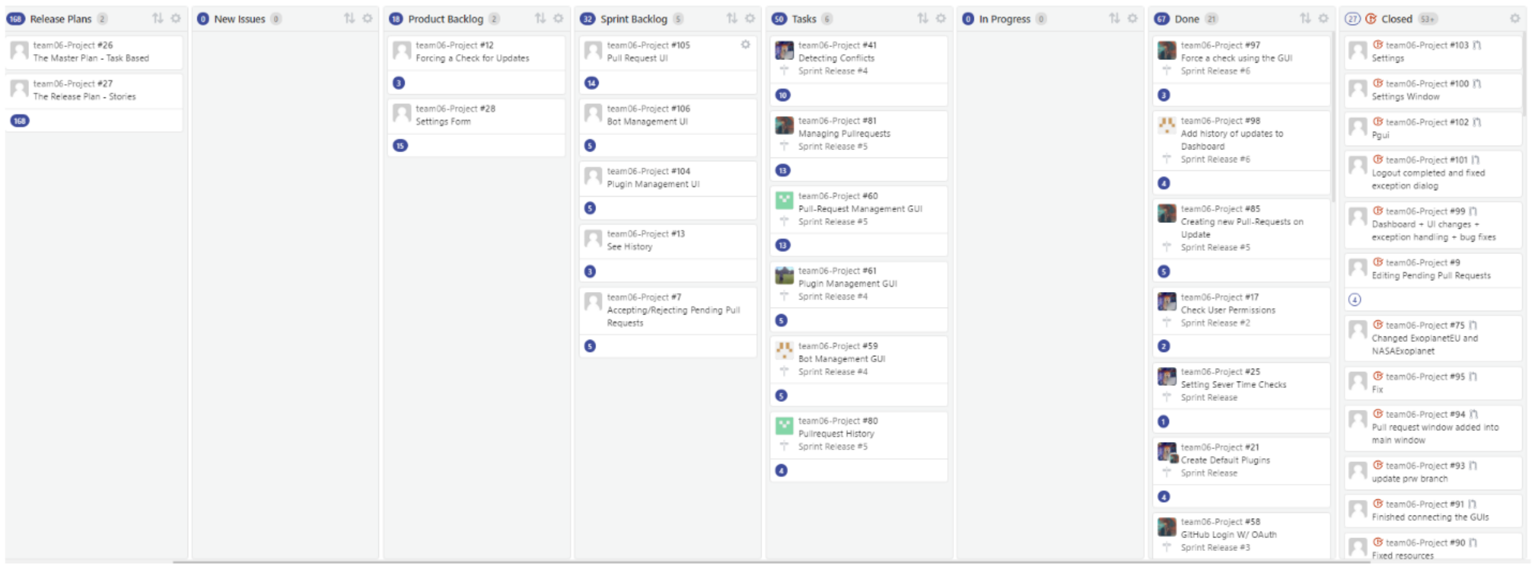
Legend:

Mark Viola
Mike Petrella
Hyeong Ook Yu
ZiXian Chen
Michael Liu

4. Sprint Results

4.1 Sprint #5 Results

4.1.1 Task Board



168 Release Plans 2
team06-Project #26
The Master Plan - Task Based
team06-Project #27
The Release Plan - Stories
168

0 New Issues 0

18 Product Backlog 2
team06-Project #12
Forcing a Check for Updates
3
team06-Project #28
Settings Form
15

32 Sprint Backlog 5
team06-Project #105
Pull Request UI
14
team06-Project #106
Bot Management UI
5
team06-Project #104
Plugin Management UI
5
team06-Project #13
See History
3
team06-Project #7
Accepting/Rejecting Pending Pull Requests
5

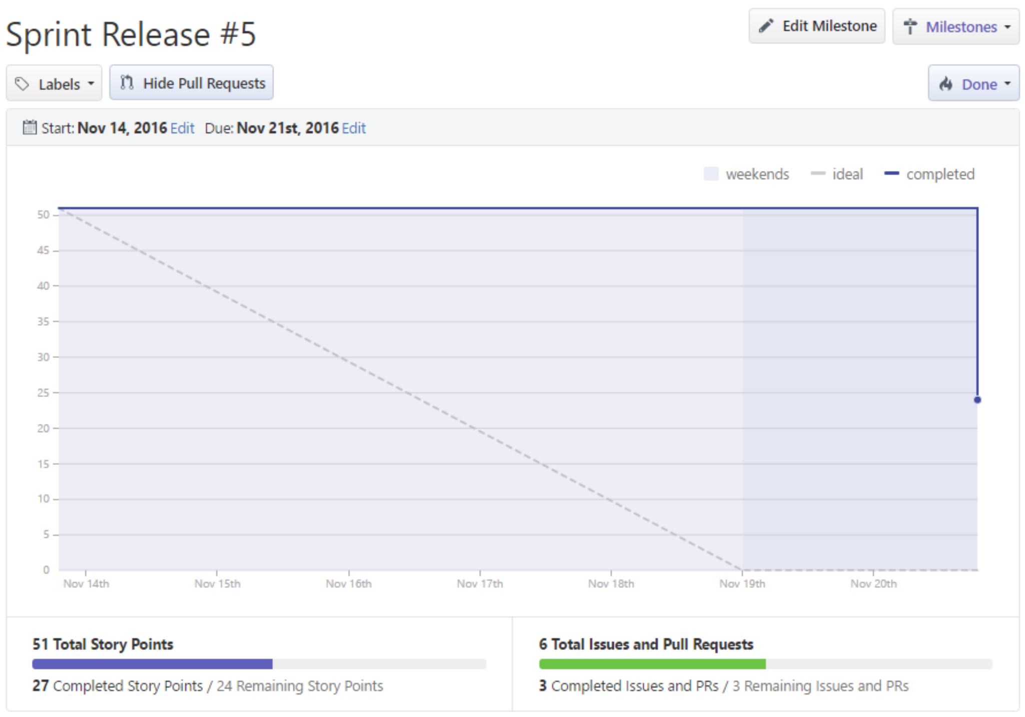
50 Tasks 6
team06-Project #41
Detecting Conflicts
Sprint Release #4
10
team06-Project #81
Managing Pullrequests
Sprint Release #5
13
team06-Project #60
Pull-Request Management GUI
Sprint Release #5
13
team06-Project #61
Plugin Management GUI
Sprint Release #4
5
team06-Project #59
Bot Management GUI
Sprint Release #4
5
team06-Project #80
Pullrequest History
Sprint Release #5
4

0 In Progress 0

67 Done 21
team06-Project #97
Force a check using the GUI
Sprint Release #6
3
team06-Project #98
Add history of updates to Dashboard
Sprint Release #6
4
team06-Project #85
Creating new Pull-Requests on Update
Sprint Release #5
5
team06-Project #17
Check User Permissions
Sprint Release #2
2
team06-Project #25
Setting Sever Time Checks
Sprint Release
1
team06-Project #21
Create Default Plugins
Sprint Release
4
team06-Project #58
GitHub Login W/ OAuth
Sprint Release #3

27 Closed 53+
team06-Project #103
Settings
team06-Project #100
Settings Window
team06-Project #102
Pgui
team06-Project #101
Logout completed and fixed exception dialog
team06-Project #99
Dashboard + UI changes + exception handling + bug fixes
team06-Project #9
Editing Pending Pull Requests
4
team06-Project #75
Changed ExoplanetEU and NASAExoplanet
team06-Project #95
Fix
team06-Project #94
Pull request window added into main window
team06-Project #93
update prw branch
team06-Project #91
Finished connecting the GUIs
team06-Project #90
Fixed resources

4.1.2 Burndown Chart



Start: Nov 14, 2016 Edit Due: Nov 21st, 2016 Edit

51 Total Story Points
27 Completed Story Points / 24 Remaining Story Points

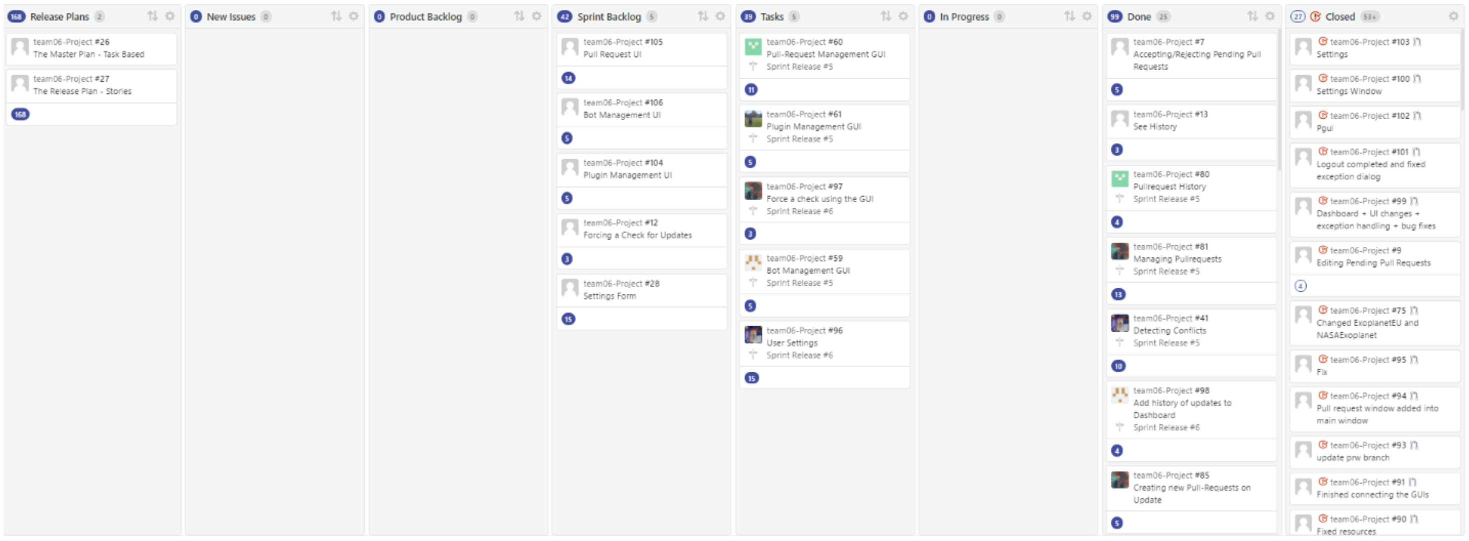
6 Total Issues and Pull Requests
3 Completed Issues and PRs / 3 Remaining Issues and PRs

4.1.3 Project Velocity

Projected project velocity: 51 story points
Actual project velocity: 27 story points

4.2 Sprint #6 Results

4.2.1 Task Board



168 Release Plans 2
- team06-Project #26
  The Master Plan - Task Based
- team06-Project #27
  The Release Plan - Stories
  168

0 New Issues 0

0 Product Backlog 0

42 Sprint Backlog 5
- team06-Project #105
  Pull Request UI
  14
- team06-Project #106
  Bot Management UI
  5
- team06-Project #104
  Plugin Management UI
  5
- team06-Project #12
  Forcing a Check for Updates
  3
- team06-Project #28
  Settings Form
  15

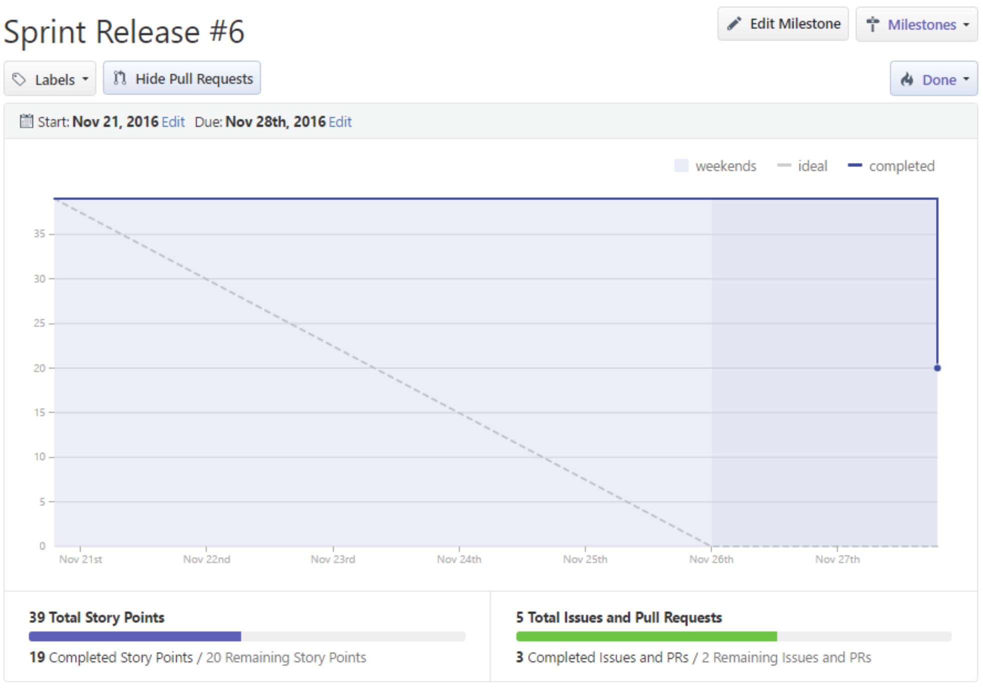
39 Tasks 5
- team06-Project #60
  Pull-Request Management GUI
  Sprint Release #5
  11
- team06-Project #61
  Plugin Management GUI
  Sprint Release #5
  5
- team06-Project #97
  Force a check using the GUI
  Sprint Release #6
  3
- team06-Project #59
  Bot Management GUI
  Sprint Release #5
  5
- team06-Project #96
  User Settings
  Sprint Release #6
  15

0 In Progress 0

99 Done 25
- team06-Project #7
  Accepting/Rejecting Pending Pull Requests
  5
- team06-Project #13
  See History
  3
- team06-Project #80
  Pullrequest History
  Sprint Release #5
  4
- team06-Project #81
  Managing Pullrequests
  Sprint Release #5
  13
- team06-Project #41
  Detecting Conflicts
  Sprint Release #5
  10
- team06-Project #98
  Add history of updates to Dashboard
  Sprint Release #6
  4
- team06-Project #85
  Creating new Pull-Requests on Update
  5

27 Closed 53+
- team06-Project #103
  Settings
- team06-Project #100
  Settings Window
- team06-Project #102
  Pgui
- team06-Project #101
  Logout completed and fixed exception dialog
- team06-Project #99
  Dashboard + UI changes + exception handling + bug fixes
- team06-Project #9
  Editing Pending Pull Requests
  4
- team06-Project #75
  Changed ExoplanetEU and NASAExoplanet
- team06-Project #95
  Fix
- team06-Project #94
  Pull request window added into main window
- team06-Project #93
  update prw branch
- team06-Project #91
  Finished connecting the GUIs
- team06-Project #90
  Fixed resources

4.2.2 Burndown Chart



Start: Nov 21, 2016 Edit Due: Nov 28th, 2016 Edit

39 Total Story Points
19 Completed Story Points / 20 Remaining Story Points

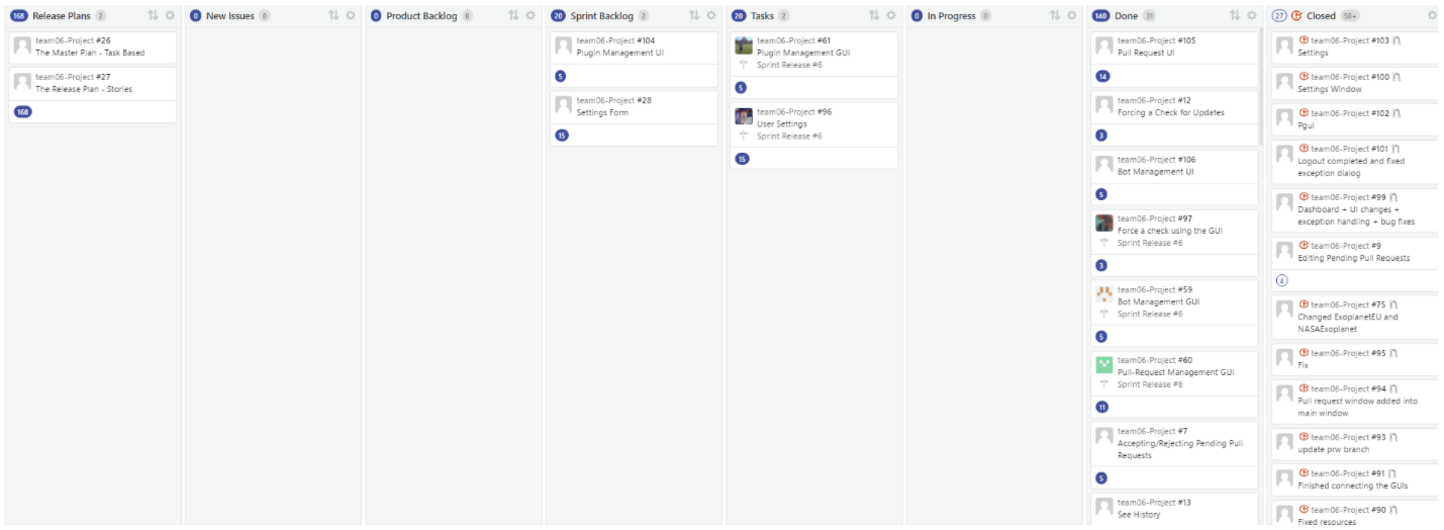
5 Total Issues and Pull Requests
3 Completed Issues and PRs / 2 Remaining Issues and PRs

4.2.3 Project Velocity

Projected project velocity: 39 story points
Actual project velocity: 19 story points

4.3 Sprint #7 Results

4.3.1 Task Board



168 Release Plans 2

team06-Project #26
The Master Plan - Task Based

team06-Project #27
The Release Plan - Stories

168

0 New Issues 0

0 Product Backlog 0

20 Sprint Backlog 2

team06-Project #104
Plugin Management UI

5

team06-Project #28
Settings Form

15

20 Tasks 2

team06-Project #61
Plugin Management GUI
Sprint Release #6

5

team06-Project #96
User Settings
Sprint Release #6

15

0 In Progress 0

140 Done 31

team06-Project #105
Pull Request UI

14

team06-Project #12
Forcing a Check for Updates

3

team06-Project #106
Bot Management UI

5

team06-Project #97
Force a check using the GUI
Sprint Release #6

3

team06-Project #59
Bot Management GUI
Sprint Release #6

5

team06-Project #60
Pull-Request Management GUI
Sprint Release #6

11

team06-Project #7
Accepting/Rejecting Pending Pull Requests

5

team06-Project #13
See History

27 Closed 50+

team06-Project #103
Settings

team06-Project #100
Settings Window

team06-Project #102
Pgui

team06-Project #101
Logout completed and fixed exception dialog

team06-Project #99
Dashboard + UI changes + exception handling + bug fixes

team06-Project #9
Editing Pending Pull Requests

4

team06-Project #75
Changed ExoplanetEU and NASAExoplanet

team06-Project #95
Fix

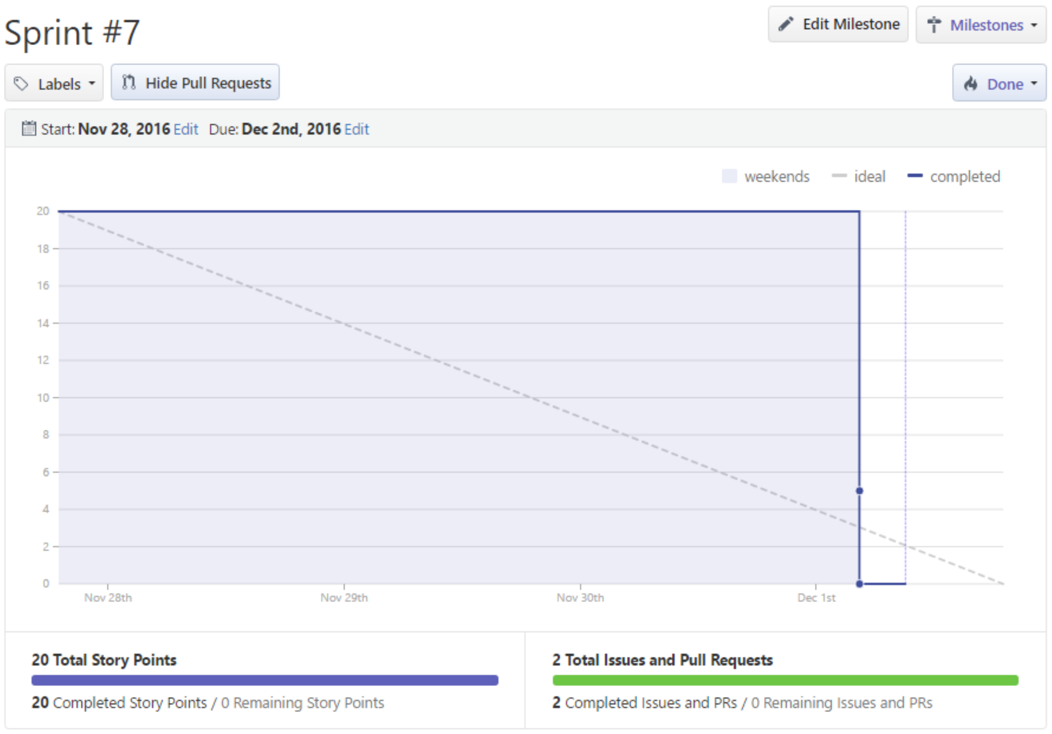
team06-Project #94
Pull request window added into main window

team06-Project #93
update prw branch

team06-Project #91
Finished connecting the GUIs

team06-Project #90
Fixed resources

4.3.2 Burndown Chart



Start: **Nov 28, 2016** Edit Due: **Dec 2nd, 2016** Edit

20 Total Story Points
20 Completed Story Points / 0 Remaining Story Points

2 Total Issues and Pull Requests
2 Completed Issues and PRs / 0 Remaining Issues and PRs

4.3.3 Project Velocity

Projected project velocity: 20 story points
Actual project velocity: 20 story points

5. System Design

5.1 Session

Manages user GitHub session. A wrapper for Octokit GitHub client, handles authorization.

5.2 Repository Manager

Manages all repository activities for a specified repo in addition of a Session. Wrapper for Octokit repository clients, deals with functions such as adding/updating files, managing branches, commits, and pull requests. Uses session as authorization for GitHub API.

5.3 Serializer

Handles plugin loading.

5.4 OECBot

Functional automation of OEC database updates. Schedule checks for plugins, and creates new updates for the OEC database, through branching, commits, and pull requests. Also uses Session to deal with GitHub API authorization. Uses PlanetMerger to handle merging of systems before committing it to OEC. Uses XMLDeserializer to parse xml files from OEC so that it may be used for comparisons. Uses serializer to get plugins.

5.5 Plugin Interface

Interface for creation and usage of plugins. Plugins are used to extract data from exoplanet databases.

5.6 CallBackServer

Utility class. Used for Oauth authorization callback, in order to obtain authorization codes. Creates Session object from GitHub callback.

5.7 HttpRequests

Utility class. Used to facilitate http requests and get back information. Plugins use this to create http requests to extract data from databases.

5.8 PlanetMerger

Utility class. Used to merge exoplanet systems from different sources.

5.9 XMLDeserializer

Utility class. Used to create systems objects from xml input.

5.10 System Design Changes

One change we made was separating GitHub API client into two different components, one just for authorization handling and one for repository managing. This way everything is compartmentalized and each class has a dedicated usage and functionality. Also this way authorization can be independent of application's main functionality.

Added PlanetMerger, because it is important to be able to get the most accurate information from all possible sources without loss of information, as well as making it so that there are less potential conflicts from multiple commits.

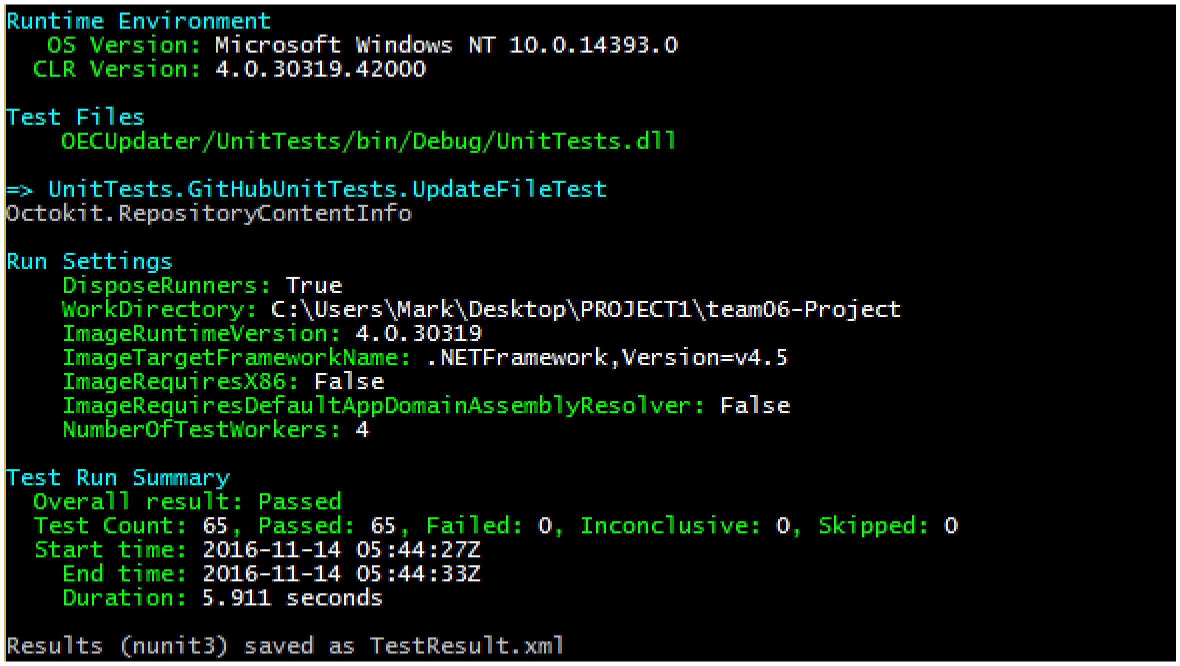
CallBackServer was added because there needed to be a way of facilitating Oauth authorization with GitHub, this way users do not need to store their password in the application and is generally more secure overall.

6. Verification

To verify that the application was meeting the specifications of Spazio's requirements, unit tests were created. The following unit tests were created:

- ExoplanetEUUnitTest
  - Unit tests for the ExoplanetEU class
- GitHubUnitTests
  - Unit tests for RepositoryManager and Session
- HTTPRequestsUnitTests
  - Unit tests for HTTPRquests
- MeasurementDSUnitTests
  - Unit tests for Measurement, MeasurementUnit, NumberErrorMeasurement, NumberMeasurement, StringMeasurement
- NASAExoplanetUnitTests
  - Unit tests for NASAExoplanetPlugin
- SerializerUnitTests
  - Unit tests for Serializer and XMLDeserializer
- StellarObjectDSUnitTests
  - Unit tests for Binary, Planet, SolarSystem, Star, and StellarObject

After running the unit tests, the following results were obtained:



```
Runtime Environment
   OS Version: Microsoft Windows NT 10.0.14393.0
  CLR Version: 4.0.30319.42000

Test Files
    OECUpdater/UnitTests/bin/Debug/UnitTests.dll

=> UnitTests.GitHubUnitTests.UpdateFileTest
Octokit.RepositoryContentInfo

Run Settings
    DisposeRunners: True
    WorkDirectory: C:\Users\Mark\Desktop\PROJECT1\team06-Project
    ImageRuntimeVersion: 4.0.30319
    ImageTargetFrameworkName: .NETFramework,Version=v4.5
    ImageRequiresX86: False
    ImageRequiresDefaultAppDomainAssemblyResolver: False
    NumberOfTestWorkers: 4

Test Run Summary
  Overall result: Passed
  Test Count: 65, Passed: 65, Failed: 0, Inconclusive: 0, Skipped: 0
  Start time: 2016-11-14 05:44:27Z
    End time: 2016-11-14 05:44:33Z
    Duration: 5.911 seconds

Results (nunit3) saved as TestResult.xml
```

65 out of 65 tests were successful which means that the software Spazio has created meets their requirements.

7. Results

7.1 Overview

Below is a summary of the result after each sprint of the project:

Sprint #1: Project fairly on track, only minor delay.
Sprint #2: Caught up with slight delay, and project on track now.
Sprint #3: Delay in project.
Sprint #4: Project halted.
Sprint #5: Greater number of developer hours assigned. Reevaluating timeline of project.
Sprint #6: Project is on track to be completed before due date.
Sprint #7: Project completed.

Proceeding sections further detail the results of sprints to highlight and explain key points of the project.

7.2 Project Velocity

The following lists the estimated project velocity throughout all the sprint iterations:

Sprint #1: 27 story points
Sprint #2: 25 story points
Sprint #3: 25 story points
Sprint #4: 25 story points
Sprint #5: 51 story points
Sprint #6: 39 story points
Sprint #7: 20 story points

The estimated project velocity was initially estimated to be ~25 story points per sprint, with the assumption that all team members would be able to consistently dedicate 5 developer hours per week. During the period of time when there was an excess of midterms and other assessments, there were issues with the amount of work completed. After Sprint #4 the trajectory of the project was reevaluated and the project velocity was increased so that completion of the project would still be possible. After Sprint #5, a majority of the core features were completed (or very close to completion) and we see that the project velocity was decreased in Sprint #6 as a result. Again, in Sprint #7 the project velocity decreased. This was due to the length of the sprint being 4 days opposed to the regular 7 days.

7.3 Adherence with Plan

Initially, the actual team results aligned with the projected plan, for Sprint #1 and Sprint #2. But during Sprint #3 and Sprint #4, the team was overwhelmed with other responsibilities so the projected plan was not properly followed. After Sprint #4, the team had an emergency meeting to re plan the project and evaluate how they could complete the application before the designated due date. The team spoke with the client again to determine if there were any features that could be taken out of the product backlog. The client stated that of the user stories in the product backlog, the ones related to displaying a history of the pull request accepts/rejects were not necessary.

Furthermore, to offset the delay in project progress, the team increased the total amount of assigned developer hours. This increase in developer hours began during Sprint #5 and allowed the project to catch up from the issues of the previous two sprints. After Sprint #6, a large majority of the project was completed and after Sprint #7 all user stories were completed. Note that although the number of completed story points on Sprint # 5 and Sprint #6, are not equal to the project velocity, this is because large tasks were very close to completion, but their values weren't burned downed since they weren't full completed.

7.4 Ending Remarks

Overall, the team was pleased with the end results of the project. Although there was a significant delay during Sprint #3 and Sprint #4, the team was able to recover during Sprint #5 and beyond, to develop the application. Upon undertaking projects in the future, the team will now take into consideration external responsibilities that team members have, and will use this to generate more appropriate sprint plans (opposed to static project velocities every sprint). This was a learning experience, and the team feels that they have learned the value of project management.