

Open Exoplanet Catalogue

Deliverable 3: Planning

By: Team 5



OEC Table of Contents

Productive Environment	4
1.1 Tool for Task Board.....	4
1.2 Tool for the Burndown Chart	4
1.3 Maintenance of Burndown Chart.....	4
1.4 Role of Each Team Member.....	4
1.5 Tools Used for Communication.....	4
1.6 Meetings In Person	4
1.7 Usage of Repository on GitHub	4
1.8 Machines Used for Development	5
Product Backlog	5
2.1 Specify Catalogues to Monitor	5
2.2 Delete Catalogue.....	5
2.3 Pull Request Notifications	5
2.4 Daily Update Check.....	6
2.5 Manually Initiate Update Check.....	6
2.6 No Duplicate Updates.....	6
2.7 Convert Data into Standard Units Used by OEC.....	6
2.8 OEC Commit Messages.....	6
2.9 Merge Updates and Changes via Pull on Git.....	6
2.10 Data Value Change Threshold.....	7
2.11 First Synchronization.....	7
2.12 Handle Alias while Updating Data	7
2.13 Handle Human Error while Updating Data	7
Release Plan.....	8
3.1 Sprint 1 (Deliverable 3)	8
3.2 Sprint 2	8
3.3 Sprint 3	8
3.4 Sprint 4 (Deliverable 4)	8
3.5 Sprint 5	9
3.6 Sprint 6 (Deliverable 5)	9
3.7 Contingency Time	9

Sprint Backlog.....	9
4.1 Download Data from Other Catalogues	9
4.2 Only Fetch Datafields In OEC.....	10
4.3 Convert Data from Catalogues into XML	10
Snapshots	11
5.1 Trello Task Board: Zoomed Out View.....	11
5.2 Trello Task Board: Normal View	12
5.3 Trello Task Board: Detailed View of Sprint Backlog.....	12
5.4 Trello Task Board: Detailed View of Sprint Task.....	13
5.5 Sprint Planner Google Sheet	13
5.6 Burndown Chart: Zoomed Out View.	14
5.7 Burndown Chart.....	14

Productive Environment

1.1 Tool for Task Board

Our team will be using Trello for the task board, combined with a Google Sheets document for both the individual sprint plan and burndown chart.

1.2 Tool for the Burndown Chart

Our team will be using a Google Sheets document viewable/writable by all team members.

1.3 Maintenance of Burndown Chart

Marhababanu (Marhaba) Chariwala will be maintaining our team's burndown chart by updating the values in the Excel table that we have set up. Each team member is expected to update the status and actual hours worked for each task they are assigned so that Marhaba knows when to update the burndown chart with completed tasks and so we have a record of actual hours worked versus estimated hours worked; so that we can revise future sprints accordingly.

1.4 Role of Each Team Member

Master Branch Maintainer: Marhababanu Chariwala

Editor: Ian Ferguson

Schedule Manager: Ahsan Zia

Trello Manager: Lucy Xing

Expert on Testing: Jubin Patel

1.5 Tools Used for Communication

A WhatsApp group we have set up will be the main medium of communication among team members. Skype or Google Hangouts will be used for online meetings. Trello and the Google Sheets Burnout file are viewable/writable by all members and will be updated in real time as tasks are completed.

1.6 Meetings In Person

Our team is planning to meet in person every Monday at 5pm and Friday at 3pm.

1.7 Usage of Repository on GitHub

Our team will be using the GitHub repository to store all finalized pdf documents for our group (in process documents are stored as google Docs accessible by all members of the group). At the end of each sprint we will also upload a copy of the burndown chart excel/Google Sheets file used for that sprint with recorded hours into a subfolder of the documents folder. Finally, we will be uploading/updating python program files as we work on them. For python files we will try to only upload files to the repository that are

currently working (although possibly non-functional) so that the current version in the repository is always a working version; however, in cases where multiple team members are working on a single file this requirement may be removed to allow collaboration (and prevent excessive merge conflicts) so long as a working version is submitted before end of sprint (with appropriate window for testing).

1.8 Machines Used for Development

- Marhababanu Chariwala –Windows laptop, home Linux computer and lab machine.
- Lucy Xing - Mac laptop, Windows laptop and lab machine.
- Ahsan Zia – Windows and Linux laptop, and lab machine.
- Jubin – Windows machine at home, and lab machine.
- Ian - Dual boot Windows 10/Linux (Ubuntu) laptop, and lab machine.

Product Backlog

Priority Scale (Low 1 - 5 High)

Cost in Story Points (1 Story Point = 1 Developer Hour)

2.1 Specify Catalogues to Monitor

Priority: 1

Cost: 3

As Prof. Cooper I would like to be able to specify URL's to be monitored for updates (specifically the Nasa Exoplanet Archive and Exoplanet.eu). I would also like to be able to monitor additional URL's in future but understand that I will need to go into the python code and provide a mapping between the data in the other catalog and the XML files in the OEC.

2.2 Delete Catalogue

Priority: 1

Cost: 2

As Prof. Cooper I would like an option to remove catalogues that are currently being monitored via a terminal command, so that they are no longer monitored for updates.

2.3 Pull Request Notifications

Priority: 5

Cost: 3

As Prof. Cooper I would like to be notified via pull requests if an update has been generated (new XML system page created) from monitored catalogues so that I can choose whether or not to update the existing catalogue.

2.4 Daily Update Check**Priority: 5****Cost: 3**

As Prof. Cooper I would like the system to be able to check for updates daily.

2.5 Manually Initiate Update Check**Priority: 3****Cost: 2**

As Prof. Cooper I would like to be able to manually initiate an update check via terminal command.

2.6 No Duplicate Updates**Priority: 4****Cost: 8**

As Prof. Cooper, I only want to be notified ONCE when discrepancies exist between other catalogues and the OEC (and the other catalog was updated more recently than the OEC); unless the other catalogue's data containing the discrepancy is updated again later.

2.7 Convert Data into Standard Units Used by OEC**Priority: 5****Cost: 5**

As Prof. Cooper, I want updates for the OEC to automatically convert data from the Nasa Exoplanet Archive and Exoplanet.eu into the standard units of measurement used by the OEC.

2.8 OEC Commit Messages**Priority: 3****Cost: 3**

As Prof. Cooper, I want updates for the OEC to contain commit messages that specify the reference URL for each update (i.e. the URL for that planet in either the Nasa Exoplanet Archive or Exoplanet.eu), and contain a hashtag identifying that the update was automatically generated by the program.

2.9 Merge Updates and Changes via Pull on Git**Priority: 5****Cost: 15**

As Prof. Cooper, I want to be able to merge the updates/changes into the OEC via a pull request on GitHub. The program should generate a separate pull request for each XML system page generated.

2.10 Data Value Change Threshold**Priority: 3****Cost: 4**

As Prof. Cooper, I want the program to not generate updates if the only differences in data values are below certain predefined thresholds (to be provided by client/specialist TA and do not include changes below the currently displayed number of decimals).

2.11 First Synchronization**Priority: 2****Cost: 5**

As Prof. Cooper, I want the first run of the program to check all planets in the other catalogues and create pull requests for all available updates for planets that have been updated more recently in the other catalogues than in the OEC. The first run should set a benchmark for future runs (i.e. monitor the dates that the planets were last updated in the other catalogues, so that no future updates are generated if the date of last update has not changed.)

2.12 Handle Alias while Updating Data**Priority: 3****Cost: 8**

As Prof. Cooper, I want the updates for existing systems (or planets) in the OEC to be XML files of the correct corresponding name/GitHub path so that a GitHub pull request will overwrite the existing file for that system and not create a file containing duplicate data with a different name. This should be done when the system (or planet) in the other catalogue is named using the same name or an alias listed in the OEC.

2.13 Handle Human Error while Updating Data**Priority: 1****Cost: 15**

As Prof. Cooper, I want the program to try and identify data in other catalogues containing possible typos/human error (hypothetical example being "Kepler-16 b" is "Kepler 16 b") and attempt to match it with the corresponding XML page in the OEC. In the given example this would mean creating a pull request for "Kepler-16 b" (OEC name) instead of "Kepler 16 b" (Nasa name).

Release Plan

Sprint Duration: 7 Days (including weekends).

3.1 Sprint 1 (Deliverable 3)

Oct 17 - Oct 23

The initial release 0.1 will contain basic program functionality (core functions upon which all future functions will be based). Release 0.1 will be able to download csv files from a given URL, and given a mapping of csv columns to XML tags will be able to generate a well formatted XML page in the style used by the OEC (1 system per page) for each exoplanet in the csv file.

User Stories Implemented:

- 4.1 Download Data from Other Catalogues
- 4.2 Only Fetch Datafields In OEC
- 4.3 Convert Data from Catalogues into XML

3.2 Sprint 2

Oct 24 - Oct 30

Release 0.2

User Stories Implemented:

- 2.9 Merge Updates and Changes via Pull on Git
- 2.3 Pull Request Notifications

3.3 Sprint 3

Oct 31 - Nov 6

Release 0.3

User Stories Implemented:

- 2.7 Convert Data into Standard Units Used by OEC
- 2.4 Daily Update Check

3.4 Sprint 4 (Deliverable 4)

Nov 7 - Nov 13

Release 0.4

User Stories Implemented:

- 2.12 Handle Alias while Updating Data
- 2.6 No Duplicate Updates

3.5 Sprint 5**Nov 14 - Nov 20**

Release 0.5

User Stories Implemented:

2.10 Data Value Change Threshold

2.5 Manually Initiate Update Check

2.8 OEC Commit Messages

3.6 Sprint 6 (Deliverable 5)**Nov 21 - Nov 27**

Release 1.0

User Stories Implemented:

2.11 First Synchronization

2.1 Specify Catalogues to Monitor

2.2 Delete Catalogue

2.13 Handle Human Error while Updating Data

3.7 Contingency Time**Nov 28 - Dec 1 (4 Days)**

If everything goes to plan the product should be complete and released at this point, however we are reserving the final 4 days of the project as contingency time in case unforeseen events cause us to need extra time. This time will also be used for outside-of-sprint tasks as mentioned on piazza post @112.

Sprint Backlog

Priority Scale (Low 1 - 5 High)**Cost in Story Points (1 Story Point = 1 Developer Hour)****4.1 Download Data from Other Catalogues****Priority: 5****Cost: 3**

As Prof. Copper, I want to be able to download csv data from other catalogues from a given URL; so that it can then be used to generate updates.

4.1.1 Implement csv downloader function using urllib library for python.**Priority: 5****Cost: 1****Completion Time: October 17, 2016****Assigned To: Marhababanu Chariwala**

4.1.2 Implement csv conversion into usable data values using csv library for python.

Priority: 5

Cost: 1

Completion Time: October 18, 2016

Assigned To: Ahsan Zia

4.1.3 Test using actual csv files from Nasa and Exoplanet.eu.

Priority: 5

Cost: 1

Completion Time: October 19, 2016

Assigned To: Ahsan Zia

4.2 Only Fetch Datafields In OEC

Priority: 5

Cost: 2

As Prof. Cooper, I want to fetch data from other catalogues only if its data field exists in the OEC (i.e. there should not be an update notification if another catalogue updated its value for “Chance of Living Organisms” in Planet X, and the OEC does not have a corresponding XML data value to “Chance of Living Organisms.”

4.2.1 Map data points from Nasa to XML values

Priority: 5

Cost: 1

Completion Time: October 20, 2016

Assigned To: Lucy Xing

4.2.2 Map data points from Exoplanet.eu to XML values

Priority: 5

Cost: 1

Completion Time: October 20, 2016

Assigned To: Lucy Xing

4.3 Convert Data from Catalogues into XML

Priority: 5

Cost: 8

As Prof. Cooper, I want to automatically generate XML system pages (one system per XML page) for the OEC for data taken from the NASA Exoplanet Archive and Exoplanet.eu.

4.3.1 Implement function taking a row of csv data and a mapping, then outputting a well formatted XML file using xml.etree.ElementTree library for python.

Priority: 5

Cost: 4

Completion Time: October 22, 2016

Assigned To: Ian Ferguson

- 4.3.2** Implement function that takes a full csv file and creates a XML file for each entry.

Priority: 5

Cost: 3

Completion Time: October 23, 2016

Assigned To: Ian Ferguson

- 4.3.3** Test using actual csv files from Nasa and Exoplanet.eu.

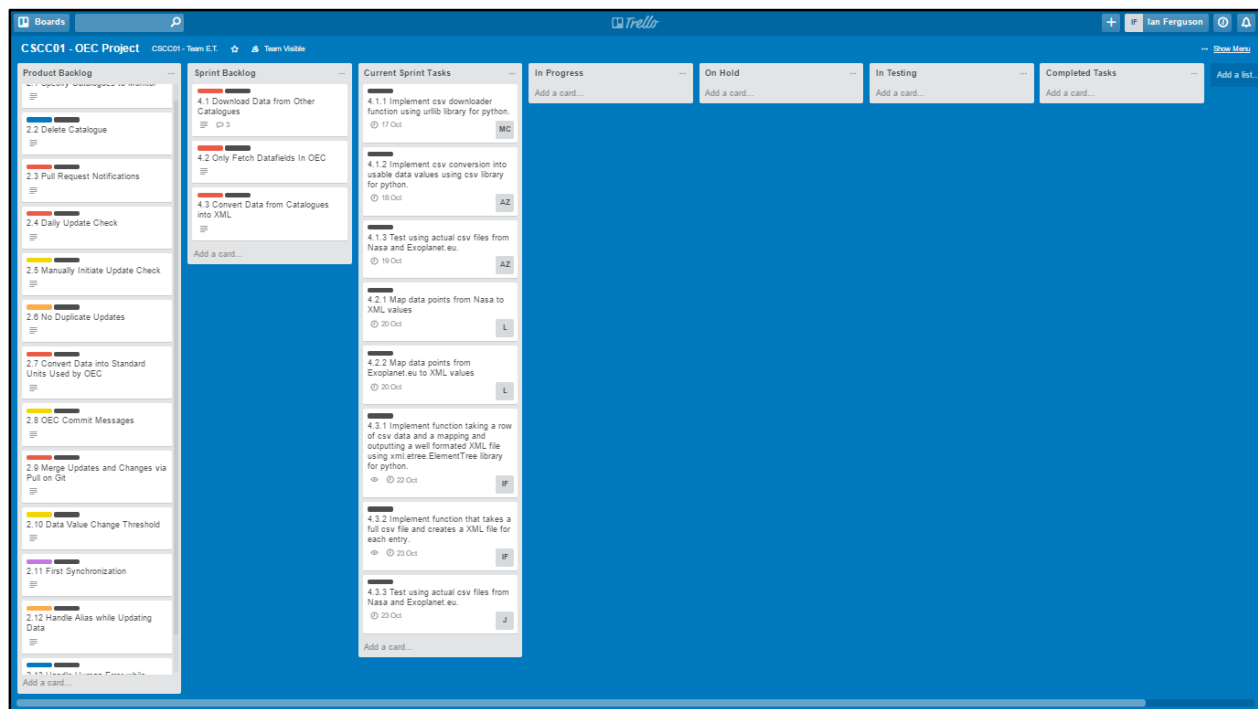
Priority: 1

Cost: 1

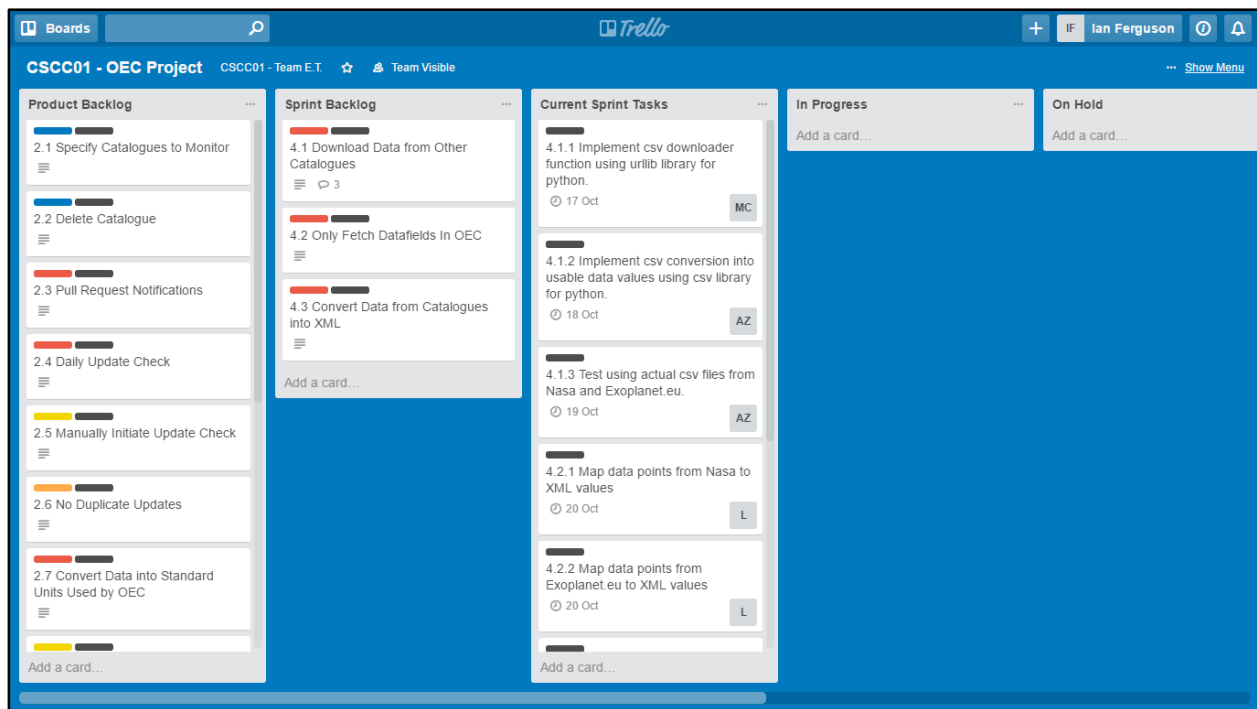
Completion Time: October 23, 2016

Assigned To: Jubin Patel

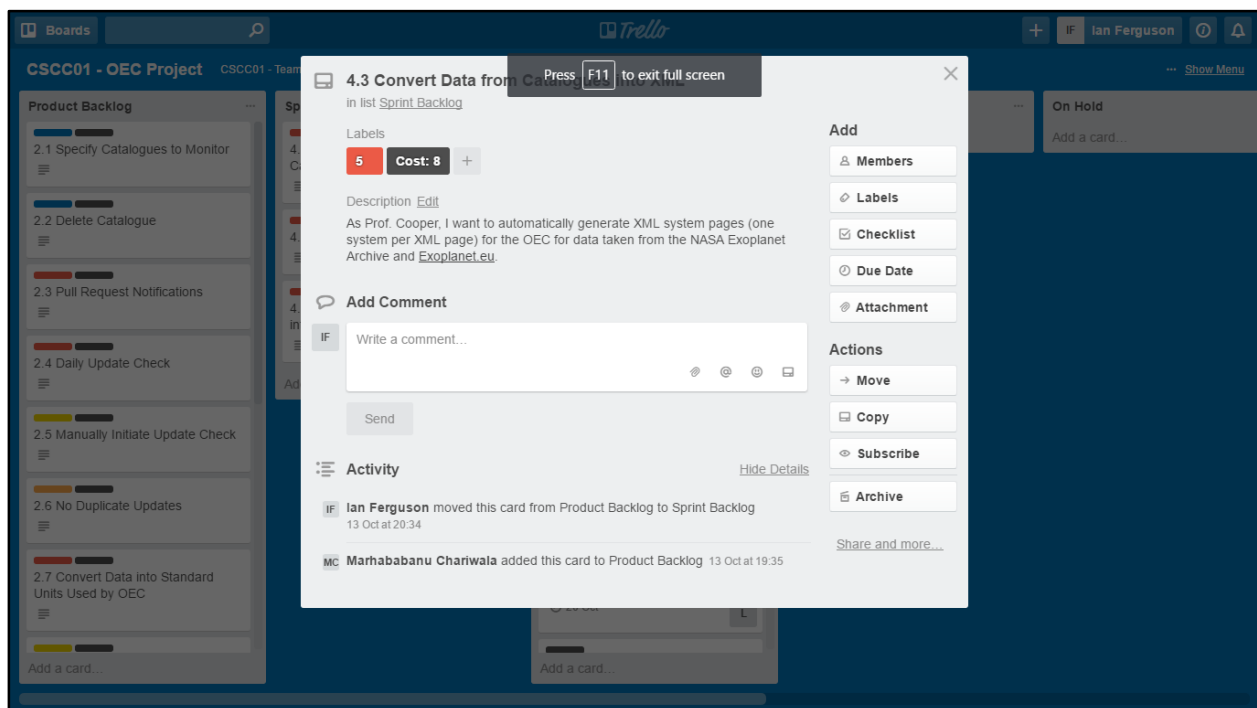
Snapshots



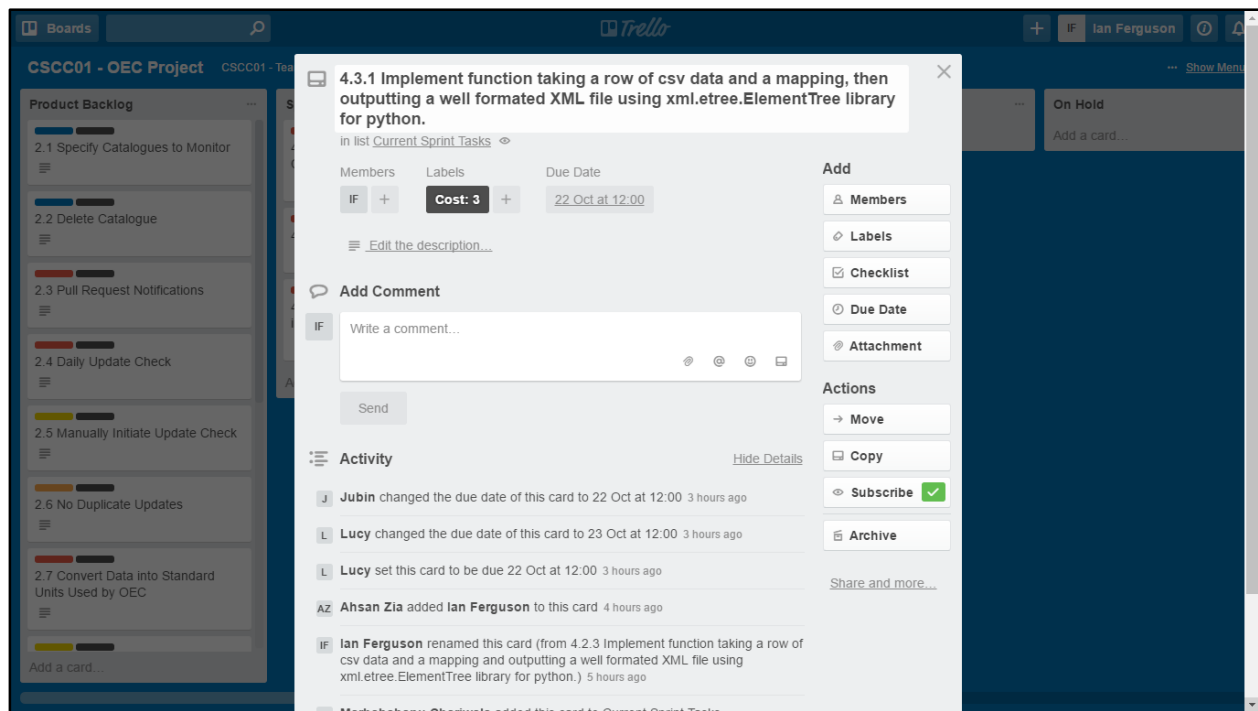
5.1 Trello Task Board Zoomed Out view: colours show priority (Red = 5, Orange = 4, Yellow = 3, Purple = 2, Blue = 1), each card has individual cost in detailed view.



5.2 Trello task board, normal view. Columns are Product Backlog, Sprint Backlog, Current Sprint Tasks, In Progress, On Hold, In Testing, and Completed Tasks.



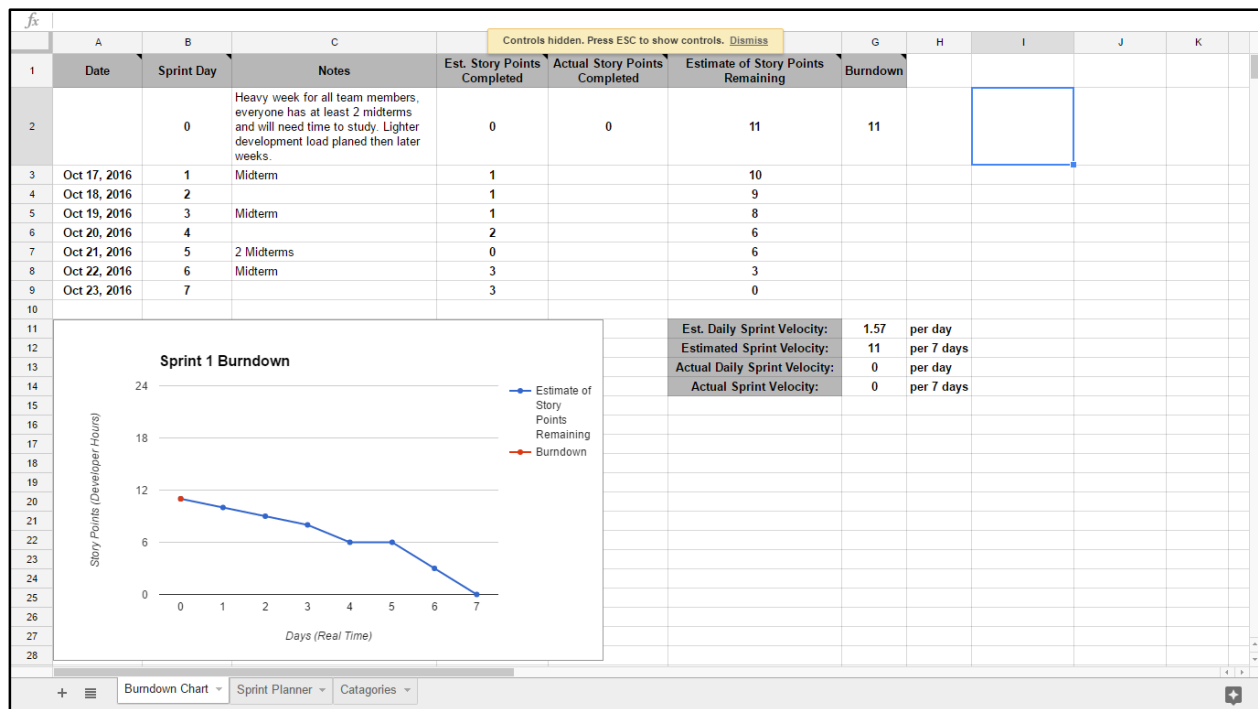
5.3 Trello task board, detailed view of Sprint Backlog User Story 4.2, Shows Priority (5), Cost (8), and detailed description. Team members may add comments, and members can be assigned to a specific User Story and or Task (no team member is currently assigned because we assign at the task, not user story level).



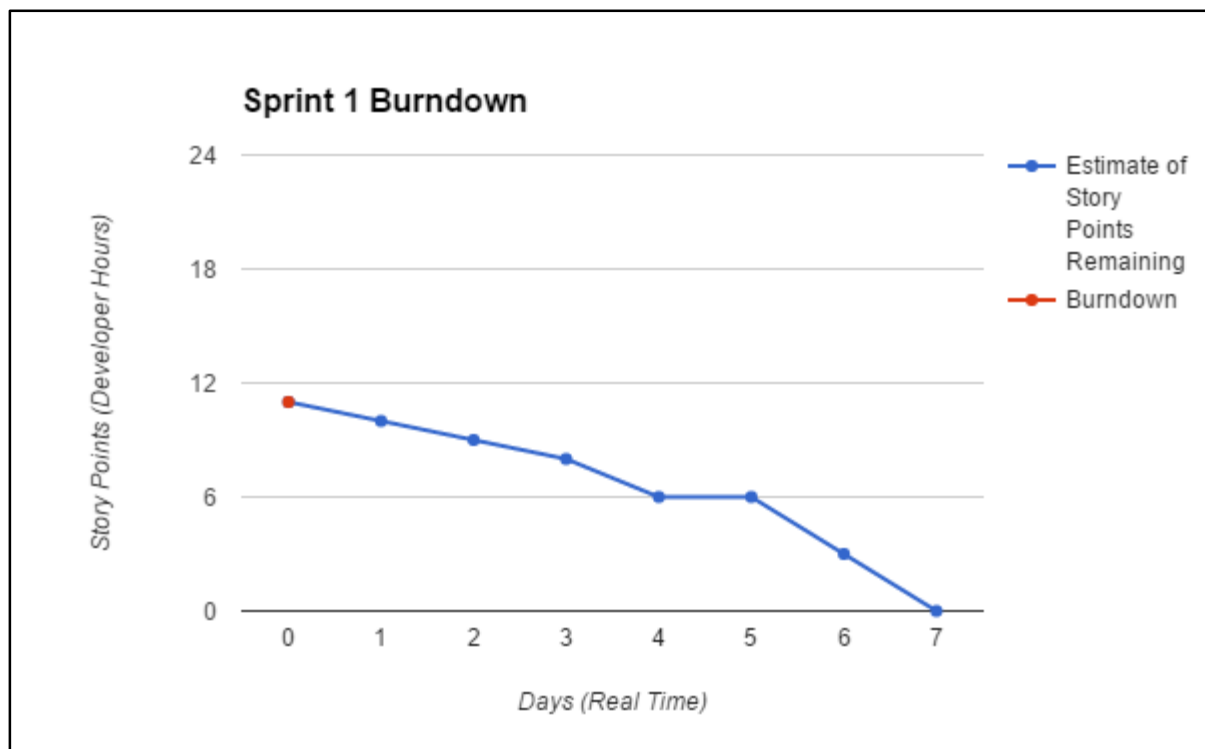
5.4 Trello task board, detailed view of Sprint Task 4.3.1, Shows Cost (3), Member assigned (Ian [IF in snapshot]), Due Date (Oct 22) and detailed description. Team members may add comments as needed.

fx	Test using actual csv files from Nasa and Exoplanet.eu.																	
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	
1	CSCC01 OEC Project Team E.T. Sprint Backlog/Burndown									Day								
2	Priority	Story Points	User Story	Tasks	Notes	Time Est	Time Spent	Team Member	Status	1	2	3	4	5	6	7		
3			Download Data from Other Catalogues: As Prof. Cooper, I want to be able to download csv data from other catalogues from a given URL, so that it can then be used to generate updates.	Implement csv downloader function using urllib library for python.		1	0	Marhaba	Not Started									
4	5	3		Implement csv conversion into useable data values using csv library for python.		1	0	Ahsan	Not Started									
5				Test using actual csv files from Nasa and Exoplanet.eu		1	0	Ahsan	Not Started									
6			Only Fetch Datafields In OEC: As Prof. Cooper, I want to fetch data from other catalogues only if its data field exists in the OEC (i.e. there should not be an update notification if another catalogue updated its value for "Chance of Living Organisms" in Planet X, and the OEC does not have a corresponding XML data value to "Chance of Living Organisms" .	Map data points from Nasa to XML values		1	0	Lucy	Not Started									
7	5	2		Map data points from Exoplanet.eu to XML values		1	0	Lucy	Not Started									
8				Implement function taking a row of csv data and a mapping and outputting a well formatted XML file using xml.etree.ElementTree library for python.		4	0	Ian	Not Started									
9	5	8	Convert Data from Catalogues into XML: As Prof. Cooper, I want to automatically generate XML system pages for the OEC for data taken from the NASA Exoplanet Archive and Exoplanet.eu.	Implement function that takes a full csv file and creates a XML file for each entry.	Will add ability to determine whether a new record SHOULD be made or not (i.e. check OEC database for existing content) in future sprint.	3	0	Ian	Not Started									
10				Test using actual csv files from Nasa and Exoplanet.eu.		1	0	Jubin	Not Started									
11							0											
+ ☰ Burndown Chart Sprint Planner Categories																		Explore

5.5 Sprint Planner Google Sheet, Custom spreadsheet used to track actual working hours for each task, has ability to assign priority, team member, time estimate per task, and recording actual time spent on each task each day (far right columns); allowing us to track and revise our time estimates in future sprints.



5.6 Burndown chart, zoomed out view. Ability to add notes (for example that we have 2 midterms on Fri so we estimate no working hours that day), estimates are filled in at beginning of sprint, and the actual story points completed is entered at the end of each day.



5.7 Burndown chart. Our burndown chart for sprint 1 with estimated daily hours added (taken before start of sprint so no burndown recorded yet).