Deliverable 5

Team 10

Vasili Skurydzin
Eric Papagiannis
Albion Fung
Tony Wu
Jerry Cheng

Table of Contents:

Product Backlog:(Current version: 3)

Priority Scale:

1 - Most Important
2 -
3 -
4 -
5 - Least Important

1 Story Point (sp) == 1 Developer Hour

~~Crossed Out~~ = Removed

USER STORIES: (Some priorities have changed since Del. 3)

1. As Hanz Hanson (Exoplanet research professor), I want to be able to initiate an update at any time which prompts the program to accumulate planet statistics information from the Open Exoplanet Catalogue on a planet-by-planet basis as one data set.
   Priority: 1, sp: 8

2. As Hanz Hanson (Exoplanet research professor),during the update I want the program to also accumulate planet statistics from the target catalogues(NASA, exoplanet.eu) as other separate data sets, including only the information fields that are present in Open Exoplanet Catalogue and omitting information not present there.
   Priority: 1, sp: 8

3. As Hanz Hanson (Exoplanet research professor), I want to be able to prompt the program to show all the differences between the two data sets compiled during the last update; for every difference I want to see the following relevant information: name of the planet, name of the star system, name of the catalogue the difference originated from (NASA or exoplanet.eu), name of the field that have been modified (ex: mass), the value of that field according to the origin catalogue (NASA or exoplanet.eu), the value of that field on the Open Exoplanet Catalogue.
   Priority: 1, sp: 20

4. As Hanz Hanson (Exoplanet research professor), I want the program to convert all units in other catalogues to the units used in Open Exoplanet Catalogue when comparing and presenting numerical values.
   Priority: 2 sp: 10

5. As Hanz Hanson (Exoplanet research professor), when viewing proposed changes, I want to see them in a numbered list.
   Priority: 1, sp: 3

6. As Hanz Hanson (Exoplanet research professor), if one of the updates discovers a planet entry in the target catalogue, while a planet with the same name is not present in the Open Source Exoplanet Catalogue, I want to the program to present it to me as a newly discovered planet with the rest of the proposed changes, providing the following information: planet name, star system name, name of the catalogue and the date the entry is posted.
   Priority: 1, sp: 4

7. ~~As Hanz Hanson (Exoplanet research professor), if I suspect that a newly discovered planet is in fact an alternative name of an existing one, I want an option to associate this planet with an existing entry in Open Exoplanet Catalogue for the future updates; the proposed addition will be postponed for the current session.~~
   ~~Priority: 3, sp: 5~~

8. As Hanz Hanson (Exoplanet research professor), I want to be able to “accept” any single proposed change presented (including added planets), prompting the program to update the information in the Open Exoplanet Catalogue by sending a single pull request to the OEC github database, containing the modified version of that same planet entry, with the field modified containing the updated value (the value that was different in the target catalogue), and all other fields remaining unchanged (after change has been accepted, it is deleted and not presented again).
   Priority: 1, sp: 15

9. As Hanz Hanson (Exoplanet research professor), I want to be able to “decline” any single change presented (including added planets), prompting the program to delete this single change after adding it to the “blacklist”, meaning same exact change will not be presented to me after future updates.
   Priority: 1, sp: 8

10. As Hanz Hanson (Exoplanet research professor), I want to be able to “postpone” any single change (including added planets), causing the program to delete this change but not add it to the “blacklist”, so that this same exact change will show up after a future update.
    Priority: 2, sp: 4

11. As Hanz Hanson (Exoplanet research professor), I want an option to clear the “blacklist”, which would cause the program to forget that some changes were previously declined and present them again in the future updates.
    Priority: 2, sp: 2

12. As Hanz Hanson (Exoplanet research professor), If I don't prompt to see the proposed changes immediately after the update, I want the program to store them for me to to access at a later time.
    Priority: 4, sp: 4

13. As Hanz Hanson (Exoplanet research professor), I want to be able to set the program to update automatically and set the time interval between updates.
    Priority: 2, sp: 5

14. As Hanz Hanson (Exoplanet research professor), I want to be able to prompt the program for its current status and get the following information: time of the last update, current auto-update settings, number of changes pending to be reviewed.
    Priority: 2, sp: 3

15. As Hanz Hanson (Exoplanet research professor), if there are changes pending to be reviewed by the time of the next update, I want the program to automatically postpone all of them.
    Priority: 2, sp: 2

16. As David Davidson (PhD Candidate), I want an option to accept or decline or postpone all currently pending changes at once.
    Priority: 4, sp: 2

17. As David Davidson (PhD Candidate), I want an option to see any single proposed change by itself, referring to it by its number in the list, omitting all other proposed changes, so that I do not get confused.
    Priority: 2, sp: 2

18. As David Davidson (PhD Candidate), I want an option to see a specific number of proposed changes at a time (for example 10 at a time), so I do not have to scroll through a lot of extra information to find what I need.
    Priority: 3, sp: 3

19. As David Davidson (PhD Candidate), I want an option to see a detailed user manual, describing the operation of the program.
    Priority: 5, sp: 4

20. ~~As David Davidson (PhD Candidate), while I am reviewing the list of proposed changes I want an option to provide a name for a certain planet and to view / accept / decline / postpone all changes for that planet at once.~~
    ~~Priority: 5, sp: 3~~

21. ~~As David Davidson (PhD Candidate), while I am reviewing the list of proposed changes I want an option to view / accept / decline / postpone all changes to a certain star system, whose name I provide, at once.~~
    ~~Priority: 5, sp: 3~~

22. ~~As David Davidson (PhD Candidate), while I am reviewing the list of proposed changes I want an option to view / accept / decline / postpone all changes originating from one of the catalogues, whose name I provide, at once.~~
    ~~Priority: 5, sp: 3~~

23. ~~As Hanz Hanson (Exoplanet research professor), I want the program to ignore minor discrepancies between the numeric values across different catalogues, given the difference is within a set percentage tolerance, present by default.~~
    ~~Priority: 2, sp: 9~~

24. ~~As Hanz Hanson (Exoplanet research professor), I want to be able to set the tolerance of every numeric field present in the planet entries in the Open Exoplanet Catalogue to the value that I choose, on a field-by-field basis, so that the program uses the tolerance entered by me, as opposed to default one.~~
    ~~Priority: 4, sp: 8~~

25. ~~As Hanz Hanson (Exoplanet research professor), I want to be able to clear all tolerances added by me, so that the program reverts to the default tolerance value for every field.~~
    ~~Priority: 5, sp: 2~~

26. As Hanz Hanson (Exoplanet research professor), if any single planet in the Open Exoplanet Catalogue database has more than one name associated with it (more than one name field in the planet entry), I want the program to recognize that the alternative name in a different catalogue (either NASA or exoplanet.eu) refers to an existing planet in the Open Exoplanet Catalogue, as opposed to a separate planet, so that the user(me) is not prompted to create duplicate entries for planets with alternative names.
    Priority: 3, sp: 5

27. As Hanz Hanson (Exoplanet research professor), I want the program to ignore minor differences in spelling of the names of planets and stars across the different catalogues, which includes case-insensitivity as well as any discrepancies in the punctuation or whitespace(number of spaces and tabs) in the planet name.

Priority: 3, sp: 5

New Stories

28. As Hanz Hanson (Exoplanet research professor), I want the date of last update on the system xml file on the OEC github database to be updated whenever I accept a proposed change corresponding to that file.
    Priority: 2, sp: 5
29. As Hanz Hanson (Exoplanet research professor), I want the error tolerance limits in XML files on OEC github database to be updated when the numerical field it corresponds to is also updated.
    Priority: 2: sp: 7
30. As Hanz Hanson (Exoplanet research professor), I want to be able to prompt the program to see a specified number of most recent proposed changes, according to the last update date in the respective exoplanet catalogues.Priority: 2, sp: 5
31. As Hanz Hanson (Exoplanet research professor), I want to be able to change the default url address of Open Exoplanet Catalogue github repository to any url address I enter, enabling the program to send pull requests to that new address.
    Priority: 5, sp: 2
32. As Hanz Hanson (Exoplanet research professor), I want to be able to give a range argument to --accept, --postpone, --deny command, resulting in changes in the given range being affected.
    Priority 5, sp: 4

Updated Release Plan (Version 3)

The last version of the release plan is to accommodate the latest changes in product backlog, including new user stories (28 - 32), as well as deleting some of the existing stories (20 - 25). The release plan is made assuming that the team is will complete implementing all of the user stories by the end of sprint 6, and the last couple of days (November 28 to December 1) are used for improving existing code and doing the refactoring and testing, as well as improving internal and external documentation.

Sprint 1: (October 17 - October 23)
- Develop the complete System Component Design;
- Implement command line operation of the program;
- Start implementing User Stories 1, 2

Sprint 2: (October 24 - October 30)
- Fully implement and test User Stories 1, 2, 3
- Start implementing User Stories 5, 17

Sprint 3: (October 31 - November 6)
- Fully implement and test user stories: 5, 6
- Start implementing user stories 4, 8

Sprint 4: (November 7 - November 13)
- Fully implement and test stories 4, 8, 10, 12, 14, 18,

Sprint 5: (November 14 - November 20)
- Fully implement stories 7, 9, 11, 13, 15

Sprint 6: (November 21 - November 28)
- Fully implement stories 16, 19, 27, 28, 29, 30, 31, 32

Period from November 28 to December 1:
- Deployment preparation activities, including but not limited to: Integration testing, code refactoring, improvements to internal and external documentation.

Note: Sprint 1 to 4 Backlogs are in previous deliverables.

Sprint 5 Backlog: (Program Version: 3)

Please see the burndown chart document for the breakdown of tasks between team members.

User Story 9:
As Hanz Hanson (Exoplanet research professor), I want to be able to “decline” any single change presented (including added planets), prompting the program to delete this single change after adding it to the “blacklist”, meaning same exact change will not be presented to me after future updates.
Priority: 1, sp: 8

- Implement “Black List” for proposed changes in driver.py (sp: 4)
- Implement “equals” method for proposed change (sp: 2)
- Implement checking every proposed change whether it is in the black list or not on update. (sp: 2)

User Story 11:
As Hanz Hanson (Exoplanet research professor), I want an option to clear the “blacklist”, which would cause the program to forget that some changes were previously declined and present them again in the future updates.
Priority: 2, sp: 2

- Implement parsing of “clearblacklist” command. (sp: 1)
- Implement clearing blacklist in driver.py when “clearblacklist” command is given by the user. (sp: 1)

User Story 13:
As Hanz Hanson (Exoplanet research professor), I want to be able to set the program to update automatically and set the time interval between updates.
Priority: 2, sp: 5

- Create callable skeleton daemon (sp: 1)
- Modify driver to allow asynchronously calling of daemon (sp: 1)
- Integrate ability for daemon to call driver’s update every set interval (sp: 2)
- Modify driver to allow killing of daemon (sp: 1)

User Story 15:
As Hanz Hanson (Exoplanet research professor), if there are changes pending to be reviewed by the time of the next update, I want the program to automatically postpone all of them.
Priority: 2, sp: 2

- In driver.py implement checking whether the list of proposed changes is empty on update. (sp: 1)

- In driver.py implement clearing the list of proposed changes on update. (sp: 1)

User Story 16:
As David Davidson (PhD Candidate), I want an option to accept or decline or postpone all currently pending changes at once.
Priority: 4, sp: 2

- Implement parsing --acceptall / --declineall / --postponeall commands to driver.py (sp: 1)
- Implement accepting / declining / postponing all changes in the list (sp: 1)

User Story 28:
As Hanz Hanson (Exoplanet research professor), I want the date of last update on the system xml file on the OEC github database to be updated whenever I accept a proposed change corresponding to that file.
Priority: 2, sp: 5

- Retrieve system's current date when xml is being modified (sp: 1)
- Modify XML lastupdate tag when any XML file is being written to (sp: 3)
- Ensure field is modified when accept is done and ensure on pull request the modification to lastupdate went through (sp: 1)

User Story 29:
As Hanz Hanson (Exoplanet research professor), I want the error tolerance limits in XML files on OEC github database to be updated when the numerical field it corresponds to is also updated.
Priority: 2: sp: 7

- Parse OEC XML for errors in every numerical field (sp: 3)
- Parse nasa and eu CSV's for errors in every numerical field present in OEC (sp: 2)
- Modify planetary objects to store the errors separately so they don't create proposed changes when comparator handles them (sp: 1)
- Modify the XML when applying proposed change to also update the error bounds (sp: 1)

User Story 30:
As Hanz Hanson (Exoplanet research professor), I want to be able to prompt the program to see a specified number of most recent proposed changes, according to the last update date in the respective exoplanet catalogues.
Priority: 2, sp: 5

- Implement parsing --showlatest # command in driver.py (sp: 1)
- Implement sorting proposed changes by the time of last update in the respective catalogue (sp: 2)
- Implement showing n latest proposed changes in driver.py as a result of --showlatest n command, preserving the original indices in the list of proposed changes. (sp: 2)

Changes since last sprint:

Since Sprint 4, our team has made minor change to the product backlog one last time, incorporating the feedback we received from the client during the short demonstration of our product's main functionality. Firstly, we decided to add a command to show a given number of most recent discovered changes, as suggested by the client (User story 30). Apart from that we have decided that it is extremely important to update "tolerance" values in the XML files in client's github database which we added as a separate user story (Story 29), as well as the "lastupdate" field in these files (Story 28). We decided to scrap some user stories concerning some convenience features, Stories 23, 24, 25 dealing with minor discrepancies in numeric fields, Stories 20, 21 dealing with viewing all proposed changes for a single Planet, Star, System; Story 7 dealing with entering an alternative name for a known planet (something the client would be easily able to do even without our application) - we decided that these are not extremely important. Instead, we decided to focus on more crucial things: ability to deny a proposed change and never see it again, as well as autoupdate settings.

Brief Overview Of Who Does What And When (Sprint 5)

S5	PLANNED						
Story	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
9	0	0	0	0	0	0	6V + 2E
11	0	0	0	0	0	1T	1V
13	0	0	0	0	0	0	5T
15	0	0	0	0	0	0	2V
16	0	0	0	0	0	1T	1J
28	0	0	5E	0	0	0	0
29	0	0	0	7E	0	0	0
30	0	0	0	0	0	1T	4J

S5	ACTUAL						
Story	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
9	0	0	0	0	0	0	6V + 2E
11	0	0	0	0	0	1T	1T
13	0	0	0	0	0	0	2V
15	0	0	0	0	0	0	5T
16	0	0	0	0	0	1T	1J
28	0	0	5E	0	0	0	0
29	0	0	0	5E	2E	0	0
30	0	0	0	0	0	1T	4E

Sprint 6 Backlog: (Program Version: 3)

Please see the burndown chart document for the breakdown of tasks between team members.

User Story 19:
As David Davidson (PhD Candidate), I want an option to see a detailed user manual, describing the operation of the program.
Priority: 5, sp: 4

- Implement functionality to display user manual. (sp: 0.5)
- Create user manual. (sp: 3)
- Implement displaying usage string on invalid input (sp: 0.5)

User Story 27:
As Hanz Hanson (Exoplanet research professor), I want the program to ignore minor differences in spelling of the names of planets and stars across the different catalogues, which includes case-insensitivity as well as any discrepancies in the punctuation or whitespace(number of spaces and tabs) in the planet name.
Priority: 3, sp: 5

- When parsing names, clean the whitespace, formatting, any any other non alphanumerical characters as an alternative name (sp: 3)
- Add alternative name in alternative name dictionary that includes the name with minimal formatting (sp: 2)

User Story 31:
As Hanz Hanson (Exoplanet research professor), I want to be able to change the default url address of Open Exoplanet Catalogue github repository to any url address I enter, enabling the program to send pull requests to that new address.
Priority: 5, sp: 2

- Implement parsing --setrepo / --clearrepo commands. (sp: 1)
- Implement storing the repo address entered by the user in memory (sp: 0.5)
- Implement resetting address to default value. (sp: 0.5)

User Story 32:
As Hanz Hanson (Exoplanet research professor), I want to be able to give a range argument to --accept, --postpone, --deny command, resulting in changes in the given range being affected.
Priority 5, sp: 4

- Implement accepting changes with the given range (sp: 1)
- Implement declining changes with the given range (sp: 1)
- Implement postponing changes with the given range (sp: 1)
- Test range operation from shell (sp: 1)

Updated System Component Design

Please refer to the PDF document “System Component Design” inside Deliverable_5 folder for the UML diagram.

System Components And Their Description / Role:

Driver

- Driver runs the program by parsing command line arguments and calling related action that it entails

Driver_commands

- Process the action parsed by driver

apiGet

- Retrieves data from nasa and exoplanet eu through get requests

XML_data_parser

- Downloads data from the Open Exoplanet Catalogue and parses it into PlanetaryObjects

CSV_data_parser and UnitConverter:

- UnitConverter (to be referred to as UC) is a nested class of CSV_data_parser (to be referred to as CDP), sicne we are converting exoplanet.eu and NASA’s units into OEC’s, no other class need to use it.
- CDP reads a CSV file given the name of the file, the source (‘eu’ or ‘nasa’ dependent on where it was retrieved) and parses it into planet objects and star objects. It can be parsed into a list or a dictionary, dependent on which function call is selected. Desired tags can also be manually selected.
- UC takes the field, name of field and the source it came from and converts the given data to OEC’s units. Note that the unit converter does not differentiate between BJD, MJD and JD due to insufficient information provided from exoplanet.eu and NASA (the conversion is not trivial).

Planetary_obect

- An object which represents more or less an interface to store data for planetary objects, which are systems, stars, or planets,and what each child class behavior is

System

- Child class of PlanetaryObject, stores data related to systems, and has references to the stars it has

Star
- Child class of PlanetaryObject, stores data related to stars, and has a references to the system it is in, and planets it has

Planet
- Child class of PlanetaryObject, stores data related to planets, and has references to the planet it is in

Comparator
- Compares planetary objects and based on differing data, provide proposed changes

Proposed_change
- Stores the conflicting data after the result of the comparators comparisons.

storage_manager
- Stores the proposed changes and other program data on disk so it can be retrieved between commands

gitClone
- Clones repository and modifies the xml that the proposed change is in, then provides a pull request

autoupdate_daemon
- Creates subprocess that automatically updates, given the interval of time in the related command

Changes Made Since Last Deliverable:

Driver.py was split into multiple files, driver and driver_commands. This was deemed necessary as driver was getting too cluttered because it was not only handling argument parsing, but also command actions. Now those responsibilities are split into their respective files, where driver does the parsing, and driver_commands processes the command that was parsed.

Storage_manager is a new feature that was also necessary in order to store the data between updates. Prior, update was called before each command was processed (due to lack of memory storage functionality), however now the result of update is stored in memory, along with other program data. Storage_manager completed the core functionality of the software.

Autoupdate_daemon is a component responsible for initiating automatic updates - a feature personally requested by the client during the presentation. Auto-update feature is initiated from driver.py, via a special command.

Brief Overview Of Who Does What And When (Sprint 6)

S6	PLANNED						
Story	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
19	0	0	0	0	0	0	1T + 3J
27	0	0	0	0	0	0	5E
31	0	0	0	0	0	0	2V
32	0	0	0	0	0	0	1T + 3E

S6	ACTUAL						
Story	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
19	0	0	0	0	0	0	4T
27	0	0	0	0	0	0	5E
31	0	0	0	0	0	0	2V
32	0	0	0	0	0	0	1T + 3E

Snapshots:

References:

Taskboard:

Please go to this link if the Trello is unclear

https://trello.com/invite/b/wOCrfemm/7f4c47d9e326865948a20e594ae288f8/oec-updater

Burndown Chart and Iteration Plans:

Please go to this link if burndown chart is unclear

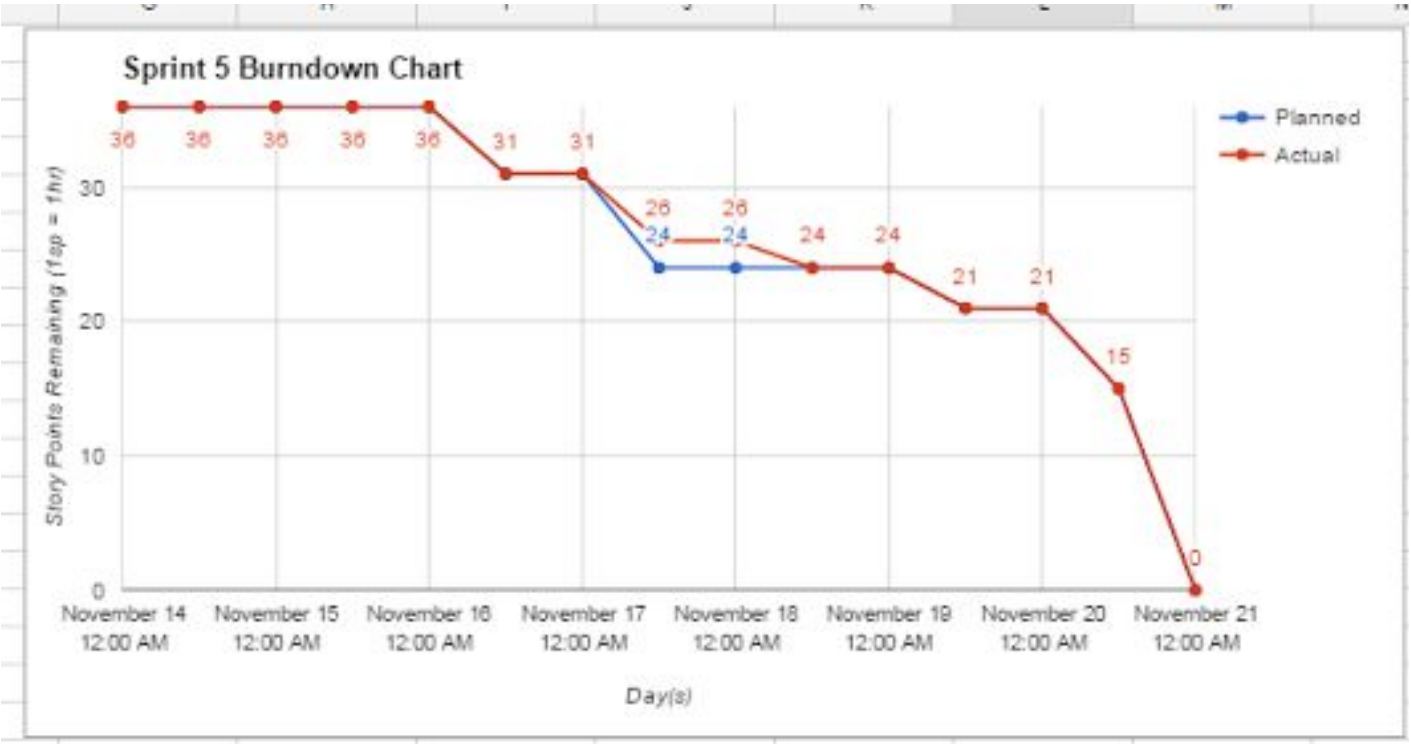
https://docs.google.com/spreadsheets/d/1D8aoThgJHv_h17ntJQxgFOzt2-ed_At3tg32SqxydJg/edit?usp=sharing

Sprint 5:

Day(s)	Planned	Actual
November 14 12:00 AM	36	36
November 14 12:00 PM	36	36
November 15 12:00 AM	36	36
November 15 12:00 PM	36	36
November 16 12:00 AM	36	36
November 16 12:00 PM	31	31
November 17 12:00 AM	31	31
November 17 12:00 PM	24	26
November 18 12:00 AM	24	26
November 18 12:00 PM	24	24
November 19 12:00 AM	24	24
November 19 12:00 PM	21	21
November 20 12:00 AM	21	21
November 20 12:00 PM	15	15
November 21 12:00 AM	0	0

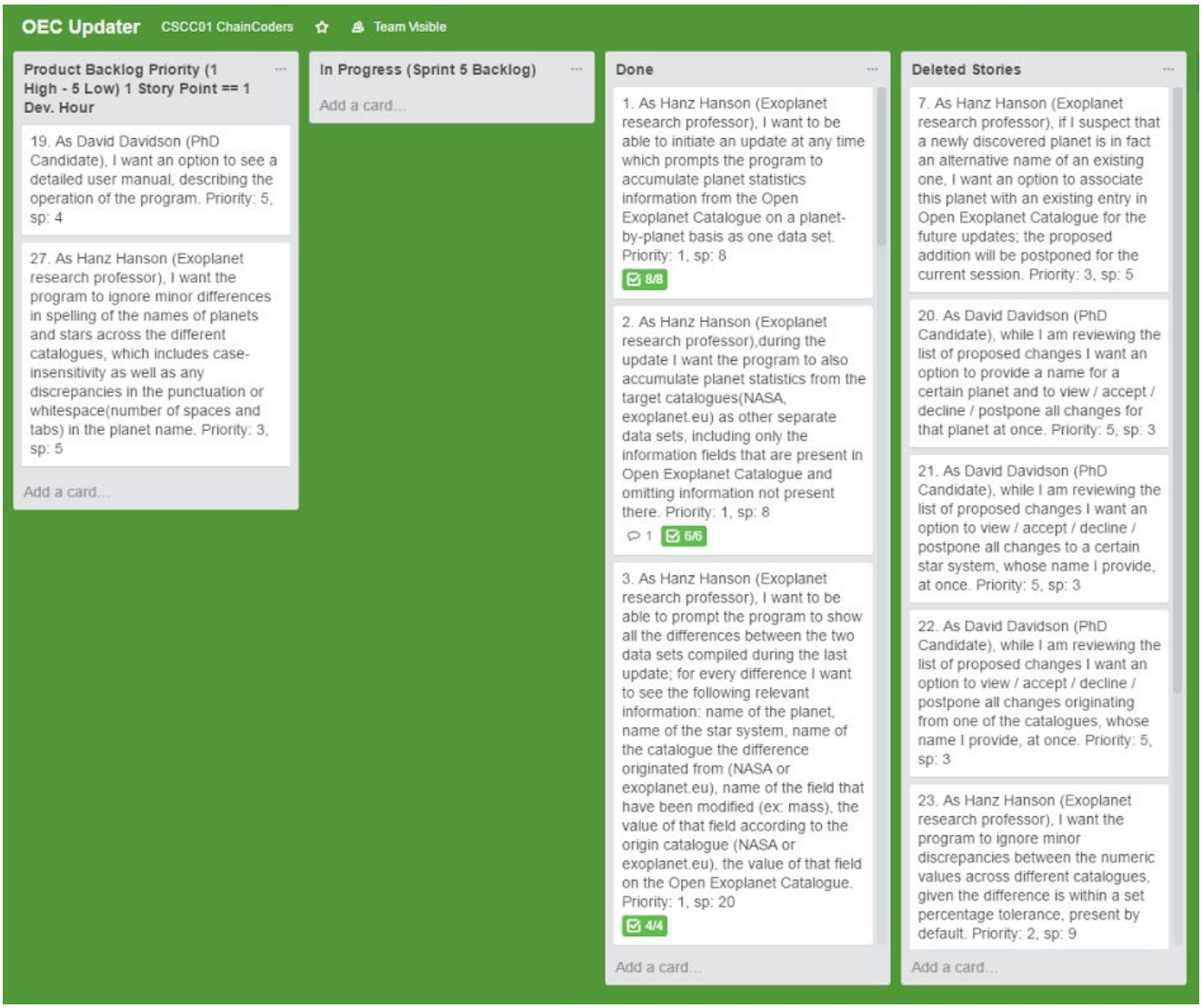
S5	PLANNED						
Story	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
9	0	0	0	0	0	0	6V + 2E
11	0	0	0	0	0	1T	1V
13	0	0	0	0	0	0	5T
15	0	0	0	0	0	0	2V
16	0	0	0	0	0	1T	1J
28	0	0	5E	0	0	0	0
29	0	0	0	7E	0	0	0
30	0	0	0	0	0	1T	4J

S5	ACTUAL						
Story	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
9	0	0	0	0	0	0	6V + 2E
11	0	0	0	0	0	1T	1T
13	0	0	0	0	0	0	5T
15	0	0	0	0	0	0	2V
16	0	0	0	0	0	1T	1J
28	0	0	5E	0	0	0	0

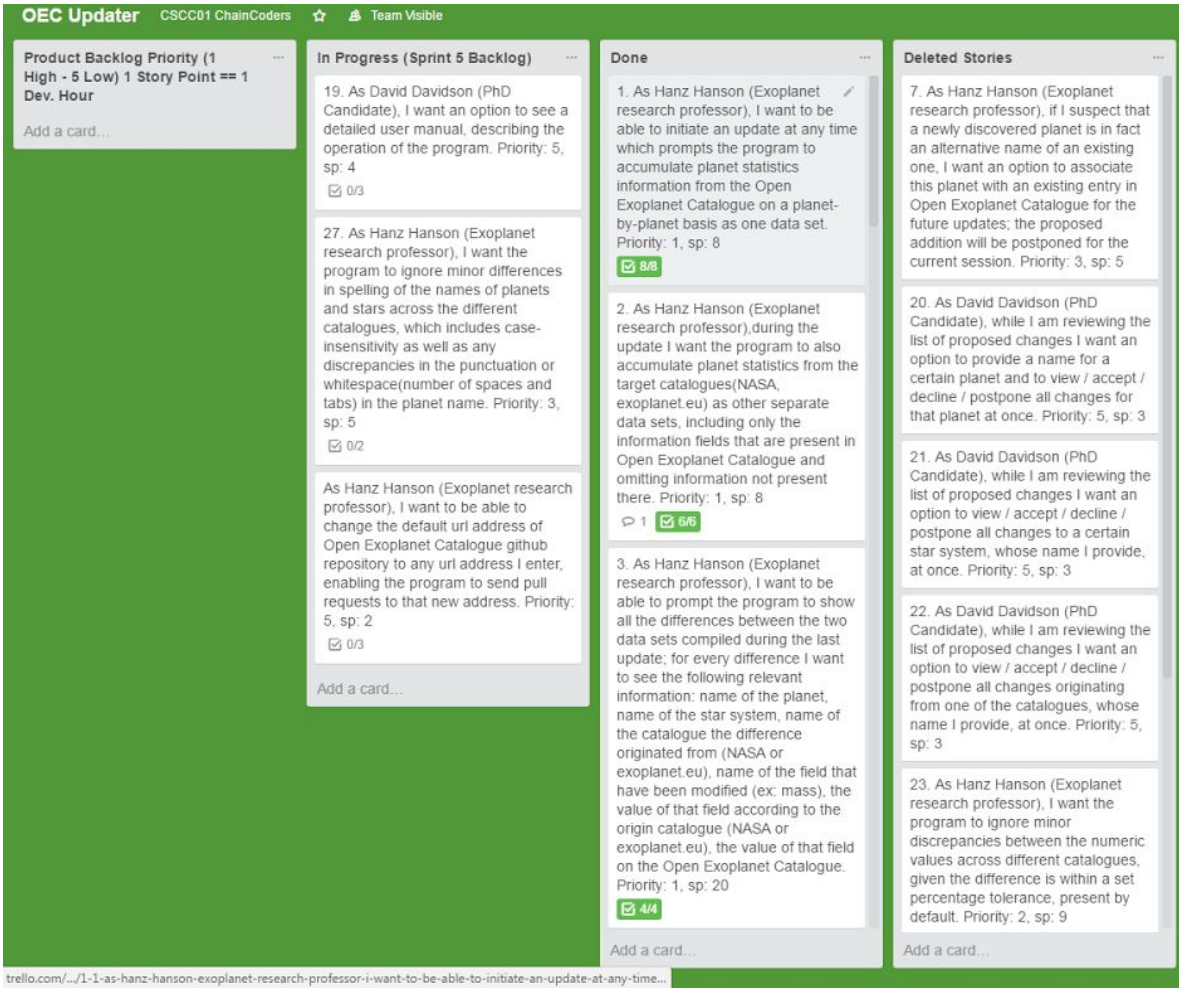


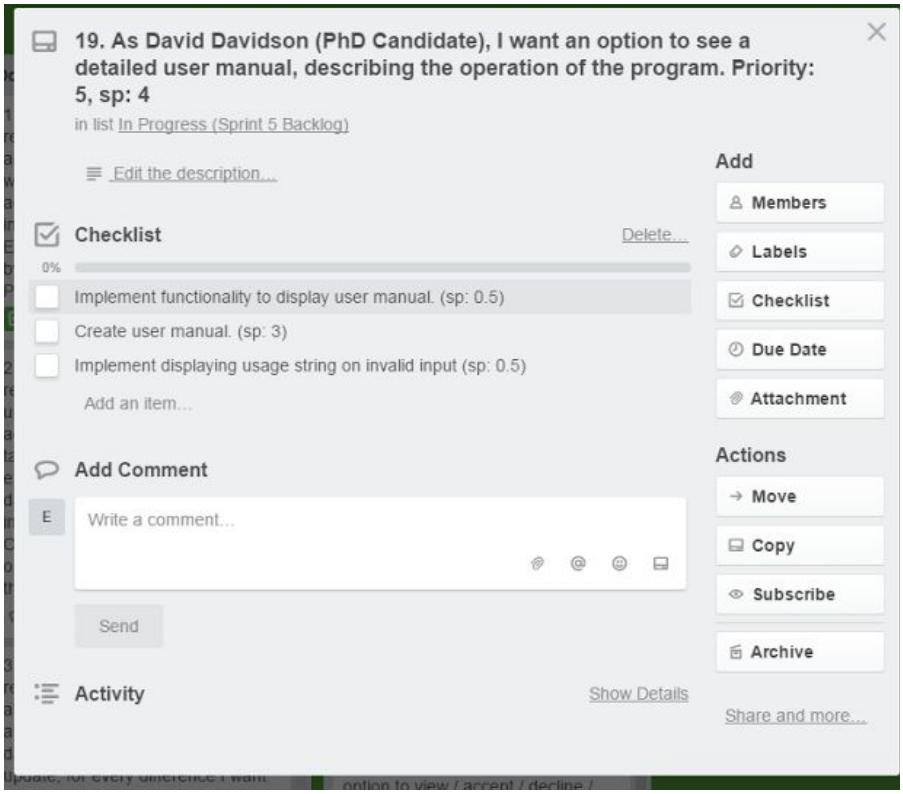
S5	PLANNED						
Story	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
9	0	0	0	0	0	0	6V + 2E
11	0	0	0	0	0	1T	1V
13	0	0	0	0	0	0	5T
15	0	0	0	0	0	0	2V
16	0	0	0	0	0	1T	1J
28	0	0	5E	0	0	0	0
29	0	0	0	7E	0	0	0
30	0	0	0	0	0	1T	4J

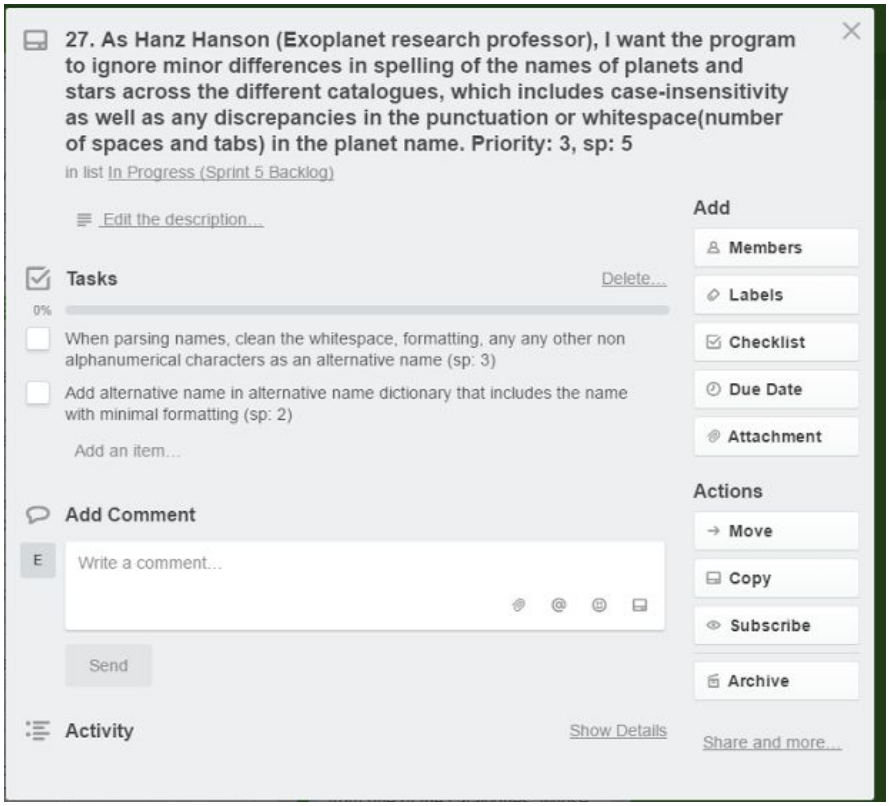
S5	ACTUAL						
Story	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
9	0	0	0	0	0	0	6V + 2E
11	0	0	0	0	0	1T	1T
13	0	0	0	0	0	0	2V
15	0	0	0	0	0	0	5T
16	0	0	0	0	0	1T	1J
28	0	0	5E	0	0	0	0
29	0	0	0	5E	2E	0	0
30	0	0	0	0	0	1T	4E

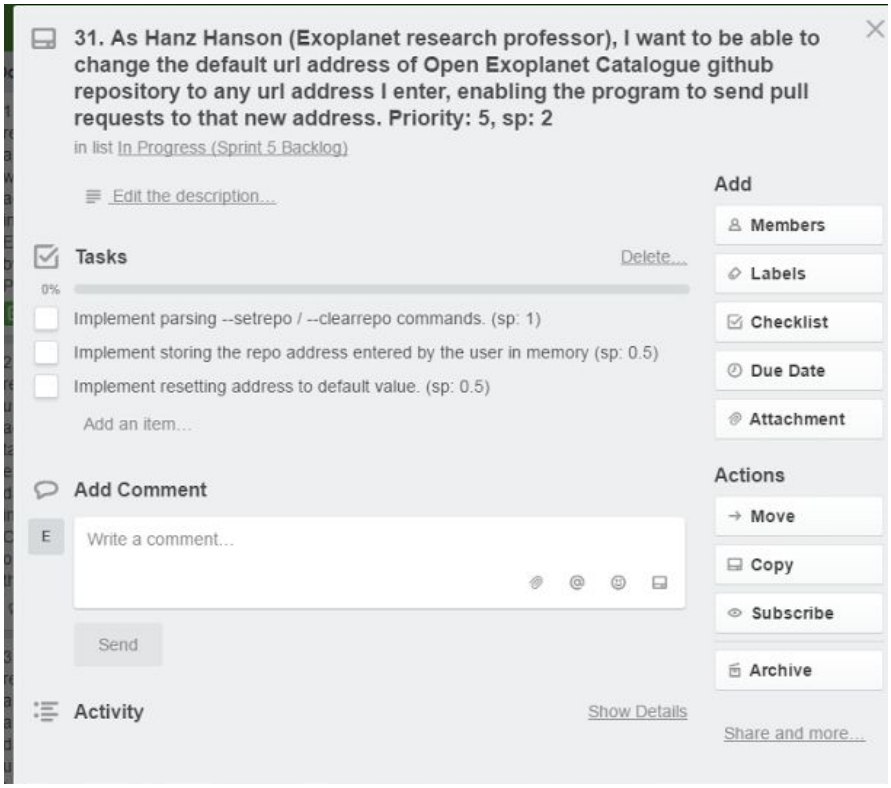

OEC Updater
CSCC01 ChainCoders
Team Visible
Product Backlog Priority (1 High - 5 Low) 1 Story Point == 1 Dev. Hour
19. As David Davidson (PhD Candidate), I want an option to see a detailed user manual, describing the operation of the program. Priority: 5, sp: 4
27. As Hanz Hanson (Exoplanet research professor), I want the program to ignore minor differences in spelling of the names of planets and stars across the different catalogues, which includes case-insensitivity as well as any discrepancies in the punctuation or whitespace(number of spaces and tabs) in the planet name. Priority: 3, sp: 5
Add a card...
In Progress (Sprint 5 Backlog)
Add a card...
Done
1. As Hanz Hanson (Exoplanet research professor), I want to be able to initiate an update at any time which prompts the program to accumulate planet statistics information from the Open Exoplanet Catalogue on a planet-by-planet basis as one data set. Priority: 1, sp: 8
8/8
2. As Hanz Hanson (Exoplanet research professor),during the update I want the program to also accumulate planet statistics from the target catalogues(NASA, exoplanet.eu) as other separate data sets, including only the information fields that are present in Open Exoplanet Catalogue and omitting information not present there. Priority: 1, sp: 8
1
6/6
3. As Hanz Hanson (Exoplanet research professor), I want to be able to prompt the program to show all the differences between the two data sets compiled during the last update; for every difference I want to see the following relevant information: name of the planet, name of the star system, name of the catalogue the difference originated from (NASA or exoplanet.eu), name of the field that have been modified (ex: mass), the value of that field according to the origin catalogue (NASA or exoplanet.eu), the value of that field on the Open Exoplanet Catalogue. Priority: 1, sp: 20
4/4
Add a card...
Deleted Stories
7. As Hanz Hanson (Exoplanet research professor), if I suspect that a newly discovered planet is in fact an alternative name of an existing one, I want an option to associate this planet with an existing entry in Open Exoplanet Catalogue for the future updates; the proposed addition will be postponed for the current session. Priority: 3, sp: 5
20. As David Davidson (PhD Candidate), while I am reviewing the list of proposed changes I want an option to provide a name for a certain planet and to view / accept / decline / postpone all changes for that planet at once. Priority: 5, sp: 3
21. As David Davidson (PhD Candidate), while I am reviewing the list of proposed changes I want an option to view / accept / decline / postpone all changes to a certain star system, whose name I provide, at once. Priority: 5, sp: 3
22. As David Davidson (PhD Candidate), while I am reviewing the list of proposed changes I want an option to view / accept / decline / postpone all changes originating from one of the catalogues, whose name I provide, at once. Priority: 5, sp: 3
23. As Hanz Hanson (Exoplanet research professor), I want the program to ignore minor discrepancies between the numeric values across different catalogues, given the difference is within a set percentage tolerance, present by default. Priority: 2, sp: 9
Add a card...

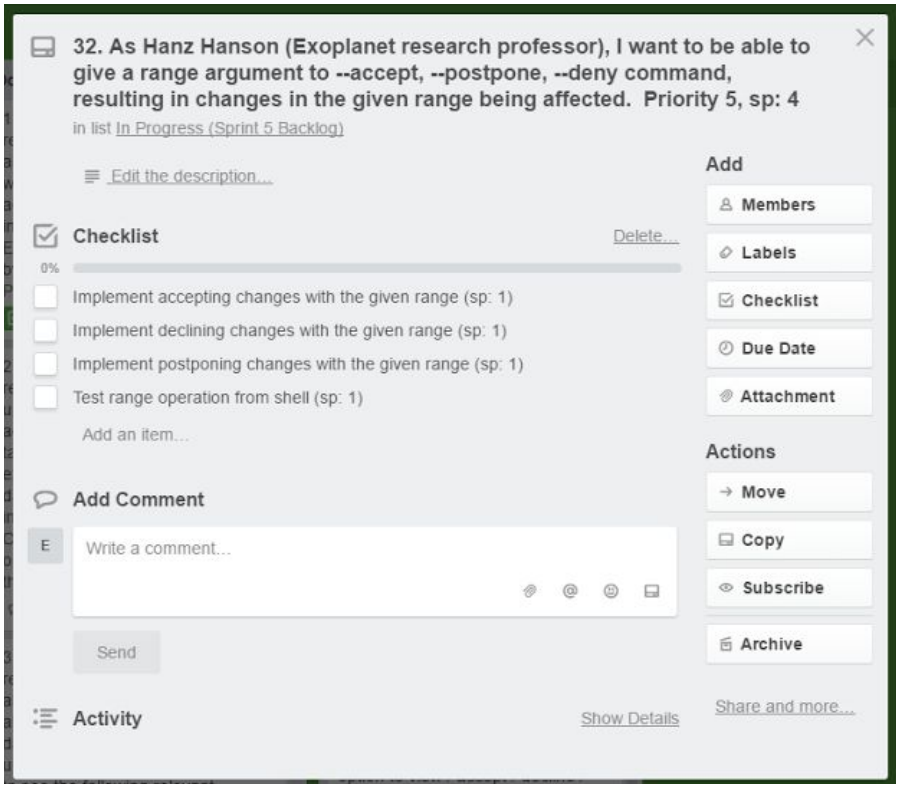
Sprint 6:
Start:



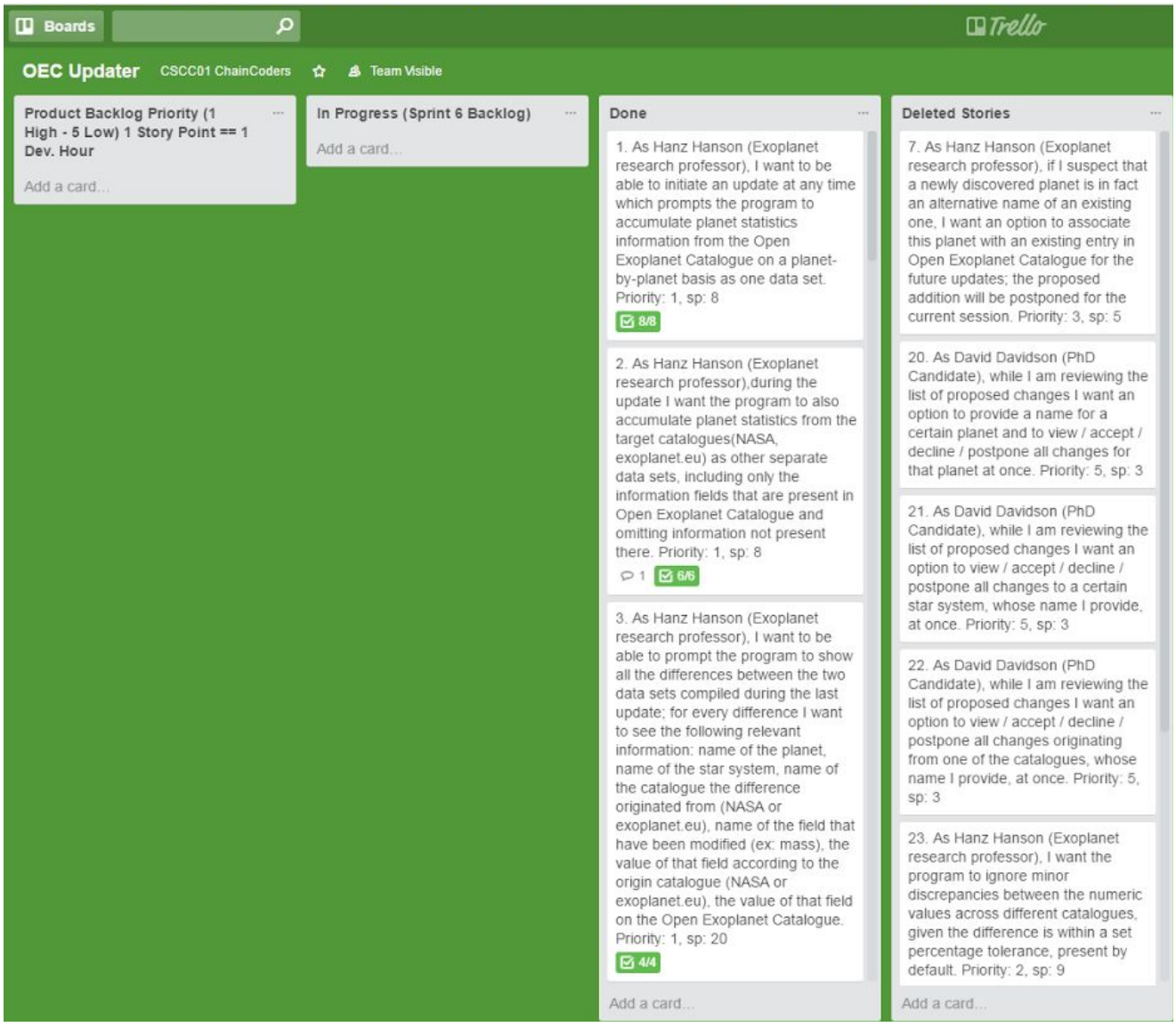

19. As David Davidson (PhD Candidate), I want an option to see a detailed user manual, describing the operation of the program. Priority: 5, sp: 4
in list In Progress (Sprint 5 Backlog)
Edit the description...
Checklist
Delete...
0%
Implement functionality to display user manual. (sp: 0.5)
Create user manual. (sp: 3)
Implement displaying usage string on invalid input (sp: 0.5)
Add an item...
Add Comment
E
Write a comment...
Send
Activity
Show Details
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more...


27. As Hanz Hanson (Exoplanet research professor), I want the program to ignore minor differences in spelling of the names of planets and stars across the different catalogues, which includes case-insensitivity as well as any discrepancies in the punctuation or whitespace(number of spaces and tabs) in the planet name. Priority: 3, sp: 5
in list In Progress (Sprint 5 Backlog)
Edit the description...
Tasks
Delete...
0%
When parsing names, clean the whitespace, formatting, any any other non alphanumerical characters as an alternative name (sp: 3)
Add alternative name in alternative name dictionary that includes the name with minimal formatting (sp: 2)
Add an item...
Add Comment
E
Write a comment...
Send
Activity
Show Details
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more...

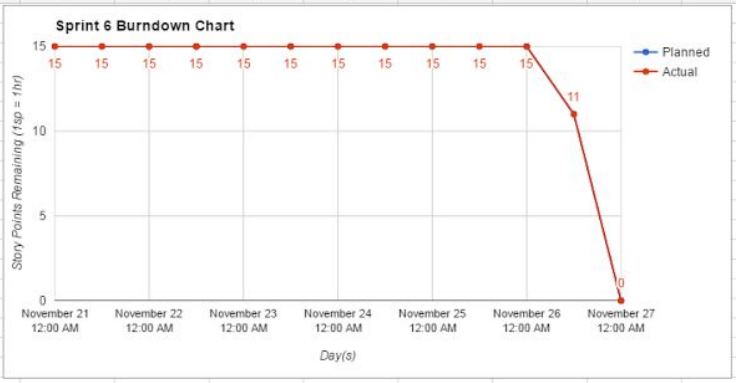

31. As Hanz Hanson (Exoplanet research professor), I want to be able to change the default url address of Open Exoplanet Catalogue github repository to any url address I enter, enabling the program to send pull requests to that new address. Priority: 5, sp: 2
in list In Progress (Sprint 5 Backlog)
Edit the description...
Tasks
Delete...
0%
Implement parsing --setrepo / --clearrepo commands. (sp: 1)
Implement storing the repo address entered by the user in memory (sp: 0.5)
Implement resetting address to default value. (sp: 0.5)
Add an item...
Add Comment
E
Write a comment...
Send
Activity
Show Details
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more...


32. As Hanz Hanson (Exoplanet research professor), I want to be able to give a range argument to --accept, --postpone, --deny command, resulting in changes in the given range being affected. Priority 5, sp: 4
in list In Progress (Sprint 5 Backlog)
Edit the description...
Checklist
Delete...
0%
Implement accepting changes with the given range (sp: 1)
Implement declining changes with the given range (sp: 1)
Implement postponing changes with the given range (sp: 1)
Test range operation from shell (sp: 1)
Add an item...
Add Comment
E
Write a comment...
Send
Activity
Show Details
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more...

End:

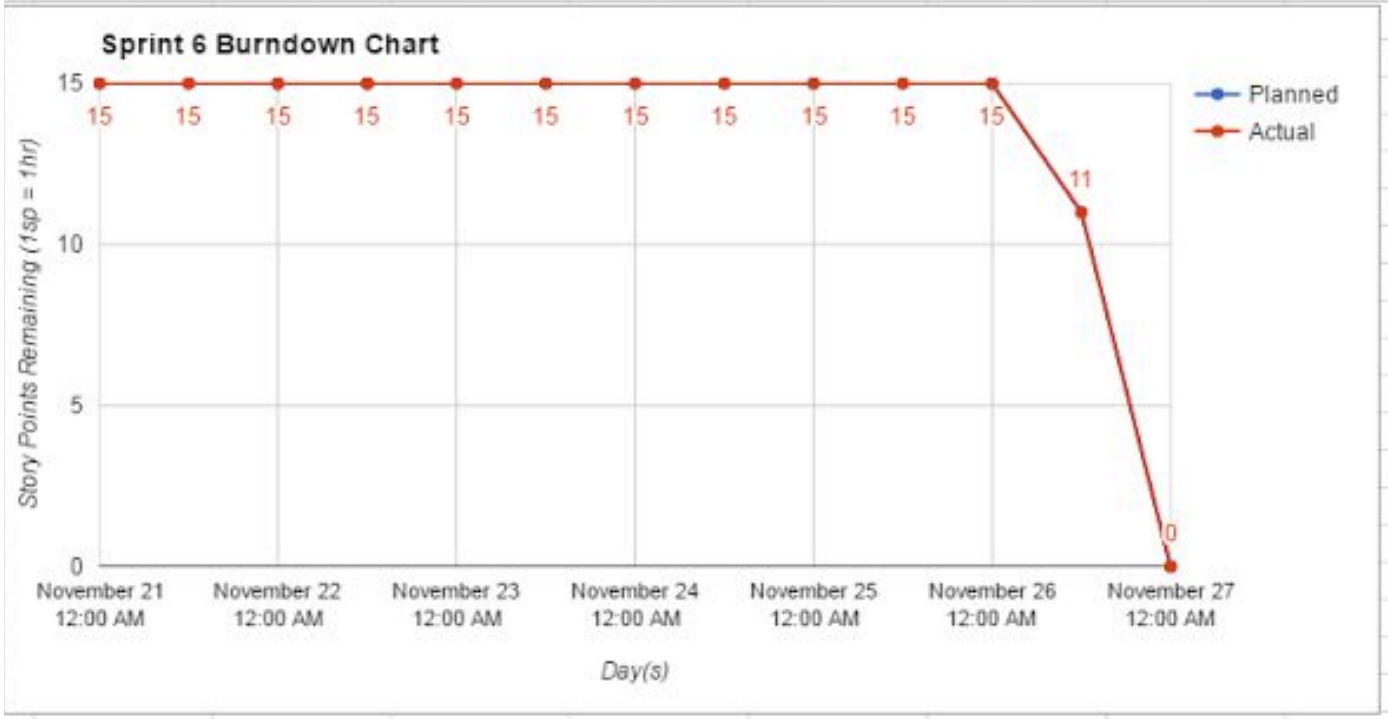

Boards
Trello
OEC Updater
CSCC01 ChainCoders
Team Visible
Product Backlog Priority (1 High - 5 Low) 1 Story Point == 1 Dev. Hour
Add a card...
In Progress (Sprint 6 Backlog)
Add a card...
Done
1. As Hanz Hanson (Exoplanet research professor), I want to be able to initiate an update at any time which prompts the program to accumulate planet statistics information from the Open Exoplanet Catalogue on a planet-by-planet basis as one data set. Priority: 1, sp: 8
8/8
2. As Hanz Hanson (Exoplanet research professor),during the update I want the program to also accumulate planet statistics from the target catalogues(NASA, exoplanet.eu) as other separate data sets, including only the information fields that are present in Open Exoplanet Catalogue and omitting information not present there. Priority: 1, sp: 8
1
6/6
3. As Hanz Hanson (Exoplanet research professor), I want to be able to prompt the program to show all the differences between the two data sets compiled during the last update; for every difference I want to see the following relevant information: name of the planet, name of the star system, name of the catalogue the difference originated from (NASA or exoplanet.eu), name of the field that have been modified (ex: mass), the value of that field according to the origin catalogue (NASA or exoplanet.eu), the value of that field on the Open Exoplanet Catalogue. Priority: 1, sp: 20
4/4
Add a card...
Deleted Stories
7. As Hanz Hanson (Exoplanet research professor), if I suspect that a newly discovered planet is in fact an alternative name of an existing one, I want an option to associate this planet with an existing entry in Open Exoplanet Catalogue for the future updates; the proposed addition will be postponed for the current session. Priority: 3, sp: 5
20. As David Davidson (PhD Candidate), while I am reviewing the list of proposed changes I want an option to provide a name for a certain planet and to view / accept / decline / postpone all changes for that planet at once. Priority: 5, sp: 3
21. As David Davidson (PhD Candidate), while I am reviewing the list of proposed changes I want an option to view / accept / decline / postpone all changes to a certain star system, whose name I provide, at once. Priority: 5, sp: 3
22. As David Davidson (PhD Candidate), while I am reviewing the list of proposed changes I want an option to view / accept / decline / postpone all changes originating from one of the catalogues, whose name I provide, at once. Priority: 5, sp: 3
23. As Hanz Hanson (Exoplanet research professor), I want the program to ignore minor discrepancies between the numeric values across different catalogues, given the difference is within a set percentage tolerance, present by default. Priority: 2, sp: 9
Add a card...

Day(s)	Planned	Actual
November 21 12:00 AM	15	15
November 21 12:00 PM	15	15
November 22 12:00 AM	15	15
November 22 12:00 PM	15	15
November 22 12:00 AM	15	15
November 22 12:00 PM	15	15
November 23 12:00 AM	15	15
November 23 12:00 PM	15	15
November 24 12:00 AM	15	15
November 24 12:00 PM	15	15
November 25 12:00 AM	15	15
November 25 12:00 PM	15	15
November 26 12:00 AM	15	15
November 26 12:00 PM	11	11
November 27 12:00 AM	0	0



S6	PLANNED						
Story	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
19	0	0	0	0	0	0	1T + 3J
27	0	0	0	0	0	0	5E
31	0	0	0	0	0	0	2V
32	0	0	0	0	0	0	1T + 3E

S6	ACTUAL						
Story	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
19	0	0	0	0	0	0	4T
27	0	0	0	0	0	0	5E
31	0	0	0	0	0	0	2V
32	0	0	0	0	0	0	1T + 3E



S6	PLANNED						
Story	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
19	0	0	0	0	0	0	1T + 3J
27	0	0	0	0	0	0	5E
31	0	0	0	0	0	0	2V
32	0	0	0	0	0	0	1T + 3E

S6	ACTUAL						
Story	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
19	0	0	0	0	0	0	4T
27	0	0	0	0	0	0	5E
31	0	0	0	0	0	0	2V
32	0	0	0	0	0	0	1T + 3E

Git Usage Strategies

Throughout the project we worked our own branches, represented by our names. This way each team member could work on their own assigned user stories without directly working on master, and also avoiding any unnecessary conflicts. The way the user stories were assigned were a reflection of how we utilized git.

Each member who worked on a feature worked exclusively on their branch, and merged to master when was it was functionally complete. Commits to master were either only merging the pull requests from the branches, uploading documentation for deliverables or extremely minor bug fixes (i.e. 1 line fixes). Major bug fixes if any were handled by the member who wrote the code in their own branch, then merged.

Description of System Validation Activities

Testing Strategies:

For this code inspection there were not a lot of new modules for us to review, that is why we focused on reviewing the new methods added in driver.py as well as other components. We were also considering things such as encapsulation, better ways to complete a task in a cleaner and more efficient manner as well as common programming paradigms and practices. For unit tests under review, we check if the test cover the majority of inputs the user can enter.

Our overall testing process can be broken down into the following:

Goal Analysis And Static Verification:

In order to validate our design, we contacted the client on multiple occasions in order to ensure our understanding of the user requirements is correct. After formulating the set of user requirements and product backlog we moved on to developing the system design that would incorporate the simplicity and basic functionality without requiring the user to read long manuals and study the intricate workings of our application.

Unit-Testing:

In order to validate our design we wrote a comprehensive test suite to test all components of our application in order to ensure every single component is functioning as intended.

Program Inspections:

Throughout the development process all members of our team were continuously checking code written by everyone else, bringing up any problems discovered along with any possible suggestions. This helped us to resolve numerous problems with compatibility due to third party libraries in a quick manner, as well as prevent further challenges and ensure team progress.

Integration Testing:

We continued our validation activities with a complete set of integration tests, created to ensure our program components, merged together work as planned and do not produce erratic or undefined behaviour.

Function Testing:

At the end of our validation activities we performed the comprehensive function testing, in order to make sure that, from the point of view of the client, the program was accomplishing its tasks, and is easy to use. We performed the shoe test very last, and our program still performed gracefully, as always.

Acceptance Testing:

In near future we would like to provide a demonstration of our application to the client, in order to make sure our design meets his requirements and to get as much feedback as we can so that we can use the time that is left to improve the functionality of our program.

New: Installation Testing:

We used VM provided to install a fresh version of the project and make sure all command are working perfectly before the final due date.

Testing Outcomes

During the unit testing phase of verification activities we have discovered a number of bugs and inconsistencies in the implementation of parsing, comparison and XML file modification, resulting in planning inconsistencies in the early stages of the project. Our team has responded by scheduling an extra meeting in order to re-plan the challenging portion of the project, making sure no all technical points are settled on, then assigning tasks to re-code the previous functionality accordingly, eliminating existing bugs.

During the unit testing we were trying to test every component in isolation, taking all corner cases into account; The output of the storage manager module test is as follows:

```
test_config_set_get_object (__main__.TestStorageManager) ... ok
test_config_set_get_simple (__main__.TestStorageManager) ... ok
test_manual (__main__.TestStorageManager) ... ok
test_read_changes_from_memory (__main__.TestStorageManager) ... ok
test_read_changes_from_memory_empty_file (__main__.TestStorageManager) ... ok
test_read_changes_from_memory_file_DNE (__main__.TestStorageManager) ... ok
test_read_empty_file (__main__.TestStorageManager) ... ok
test_read_file (__main__.TestStorageManager) ... ok
test_read_nonexistant_file (__main__.TestStorageManager) ... ok
test_write_changes_to_memory (__main__.TestStorageManager) ... ok

----------------------------------------------------------------------
Ran 11 tests in 0.005s
```

The above test suite tests storage_manager functionality, and it is written to account for all corner cases: test_read_nonexistant_file, test_read_changes_from_memory_file_DNE, test_read_changes_from_memory_empty_file, etcetera.

Similar strategy was applied to other components as well, for example, Comparator test output:

```
testCreateComparatorWithDifferentPlanetaryObjects (__main__.TestComparator) ... ok
testCreateComparatorWithNonPlanetaryObjects (__main__.TestComparator) ... ok
testInnerJoinDiffFieldDiff (__main__.TestComparator) ... ok
testInnerJoinDiffFieldMatch (__main__.TestComparator) ... ok
testSQLjoin (__main__.TestComparator) ... ok
testStarCompareEmptyStarsPlanetA (__main__.TestComparator) ... ok
testStarCompareEmptyStarsPlanetDC (__main__.TestComparator) ... ok
testStarCompareEmptyStarsPlanetDN (__main__.TestComparator) ... ok
testStarCompareEmptyStarsPlanetN (__main__.TestComparator) ... ok
testStarCompareEmptyStarsStarC (__main__.TestComparator) ... ok
testStarCompareEmptyStarsStarN (__main__.TestComparator) ... ok
testStarCompareStarWithOneFieldPlanetA (__main__.TestComparator) ... ok
testStarCompareStarWithOneFieldPlanetDC (__main__.TestComparator) ... ok
testStarCompareStarWithOneFieldPlanetDN (__main__.TestComparator) ... ok
testStarCompareStarWithOneFieldStarC (__main__.TestComparator) ... ok
testStarCompareStarWithOneFieldStarN (__main__.TestComparator) ... ok
testStarCompareStarWithOneFieldStarN2 (__main__.TestComparator) ... ok
testStarCompareWithNonStarObjects (__main__.TestComparator) ... ok
```

testproposedChangeStarCompare (__main__.TestComparator) ... ['planet1', 'planet3'] ['planet1', 'planet3'] Ok

This Test suite was created to make sure that the Comparator module works with all possible inputs, for example: two empty Planet / Star objects, two Planets in which one field differs by something insubstantial like a newline; two Planetary objects, differing in the number of fields, etcetera.

There is more of course. (Tests folder)

Testing edge cases ensured that all components work as intended in isolation, which enabled the team to move on to integrate all units together and implement all the functionality planned in product backlog.

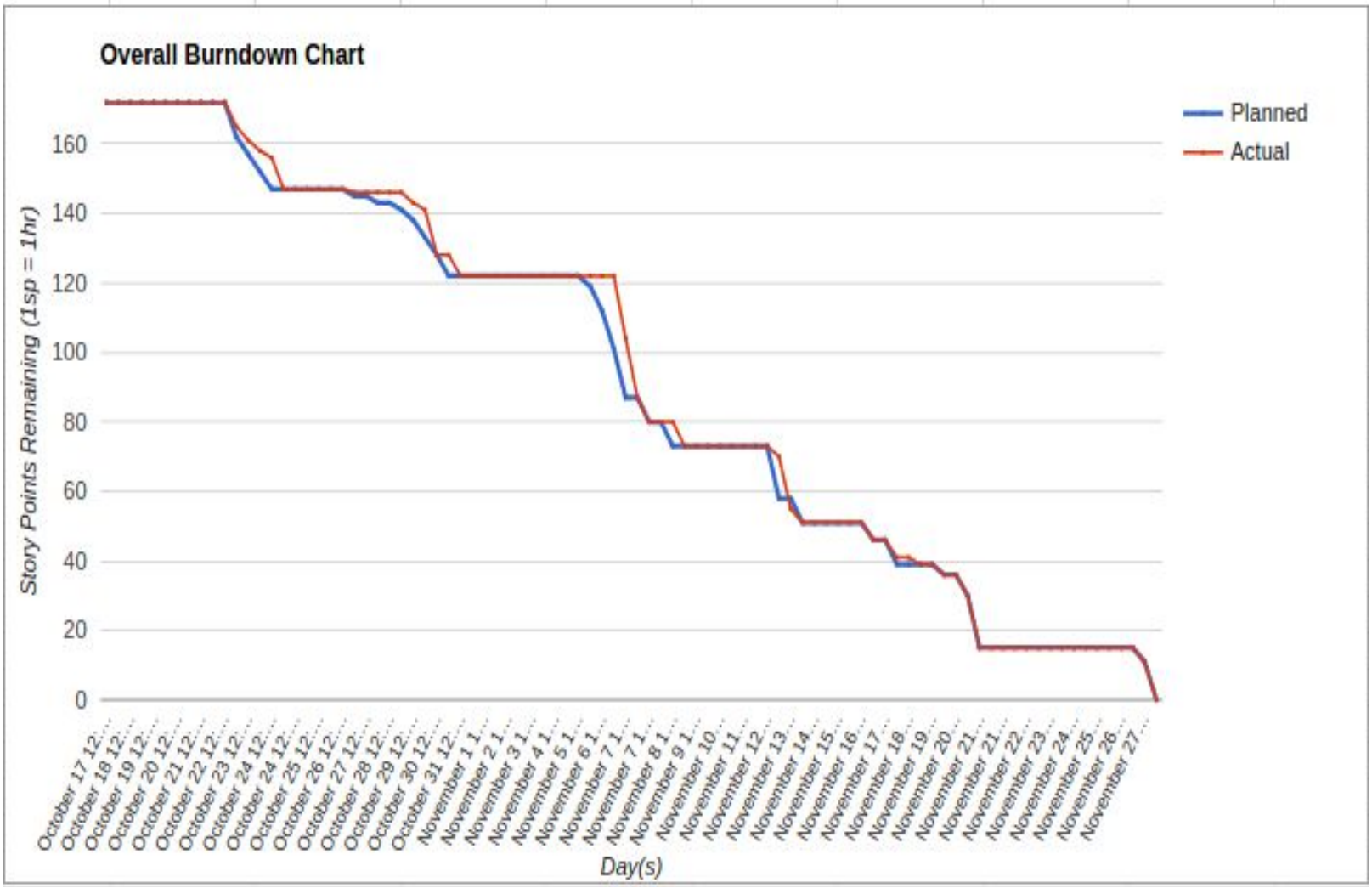
An important outcome of code inspection activities undertaken by the team was a large number of shortcomings in the external documentation in many methods and some instances of unreadable code. As a result of these findings our team has formulated a centralized strategy to keep all the documentation consistent for extension and maintenance purposes, which proven crucial in later stages of the project, while implementing user stories which had to do with displaying most recent proposed changes, as well as storing program information with memory.

We believe that without the system validation activities our project would be much harder to maintain and to add new features too; It is also quite possible that without thorough testing our project would not be as robust and tolerant to invalid user input.

You can find our tests in the “test” folder.

Burndown Chart For The Entire Project

Link to view:
https://docs.google.com/spreadsheets/d/1D8aoThgJHv_h17ntJQxgFOzt2-ed_At3tg32SqxydJg/edit?usp=sharing



Brief Overview Of The Project Saga

Throughout the project the design of the program has gone through a lot of changes. The project began with utilizing mostly UNIX software tools to complete tasks and using Python as an overlay. However, the software tools proved to be a poor fit for Python in the team's collective opinion, and the project had to be replanned in order to take advantage of more object-oriented approach.

Throughout the course of the project and development, the team was calm and collected in resolving difference of ideas such as the one mentioned previously. While the team was usually busy during the week, the team members were still able to gather for regular Wednesday meetings, where important design and planning decisions were made, and dedicate the weekends to doing the work to complete user stories, maintaining communication through Discord and text messages.

While the first sprint was more or less on schedule, the second sprint was significantly behind. However, by the fourth sprint not only was the team caught back up to schedule, but was in fact ahead of the plan. The product backlog was subjected to slight modifications regarding command line usage after Sprint 4, and remained unchanged ever since. By the end of Sprint 6 all the planned user stories were completed. As the semester nears its end, the team decided to dedicate last couple of days to refactoring and testing the existing code in a calm and relaxed atmosphere.

Project Velocity

The team had set a estimated project velocity of 25 story points per sprint, which translates to an average of 5 story points per person on the team, where each point is approximately one hour of work. On the first two sprints, the performance did not meet the estimated project velocity, where the team was severely behind on the second week due to midterms. However, on the third and fourth sprint, the team exceeded the estimated project velocity and completed more work than that was estimated can be completed. In terms of deliverables, sprint one was the first software deliverable, sprint two, three contributed to the second software deliverable, and sprint four, five and six contributed to the third software deliverable.

Planning And Replanning

The team had to replan some of the workload in order to meet the desired overall project velocity and complete all user stories and tasks as well as to accommodate for midterms and school work. This happened during Sprint 2 where for most team members three midterms

happened in the same sprint, and in sprint three in order to have the team catch up back to schedule.

In terms of software design and planning, the design was changed multiple times to accommodate what the client wanted as well as the team discovers better solutions towards problems the team had encountered. Modifications to system design happened across first three sprints as the team discussed problems such as how to retrieve data from the databases, the methodology in which difference in data will be compared, and so on.

Project planning also changed as the team developed better understanding of client requirement and the best strategy to meet them. The team ensured all its members understood and agreed to the changes in the plan.

Progress And Result Of Deliverable 5

Paperwork and planning is definitely more solid compared to the earlier deliverables. The product backlog has become more specific, while the iteration plan and burndown chart has become more accurate and correct. Since the heaviest workload had already been completed in the previous sprints, the sprints in deliverable 5 are relatively less productive due to the lack of tasks to complete (Many hours were put in into refactoring, bug fixing and documentation improvements). However, work has still been completed and new functionality has been introduced at the end of every sprint. Our team has completed all of the planned user stories and all the team members are satisfied with the final product.