Deliverable 3

Team 10

Vasili Skurydzin
Eric Papagiannis
Albion Fung
Tony Wu
Jerry Cheng

Table of Contents:

Deliverable 3 : Q & A ……... 2

Revised Personas …….. 3

Product Backlog (With Revised User Stories) ……..…….. 5

Release Plan ……...……....... 9

Sprint Backlog…… 10

Rough Plan of Who Does What and When ……..... 12

Snapshots of Setup ……..……..... 13

Answers to Deliverable 3 questions:

1. Tools for task board: Trello
   https://trello.com/b/wOCrfemm/oec-updater
2. Tools for burndown chart: Google Sheets
   https://docs.google.com/spreadsheets/d/1D8aoThgJHv_h17ntJQxgFOzt2-ed_At3tg32SqxydJg/edit?usp=sharing
3. Burndown Maintenance: Eric as admin, maintained through Google Sheets so that everyone can easily contribute
4. Roles of team members
    a. Eric: development, maintaining the burn-down chart,
    b. Vasili: development, maintaining the task board, communication with the client
    c. Albion: development, maintaining the master branch on github
    d. Jerry: development, communication with Computer Science TA
    e. Tony: development, compatibility testing - includes operation on Linux
5. Means of communication: Discord, Skype, email, texting
6. Time of in person meeting: Wednesday 1100-1300h for in person meeting, as well as during TA time
7. The github repository will contain the final version of all deliverables, as well as work in progress. The deliverables will be contained in folder “Deliverables”, which will also contain readme text file explaining the contents of the repository. No unnecessary files will be pushed into the master branch.
8. Computers:
    a. Albion:
        i. MacBook Pro
        ii. Desktop w/ Windows 10
    b. Vasili:
        i. Windows 8 laptop
    c. Jerry:
        i. Windows 10/Linux
    d. Eric:
        i. Windows 7 / Linux
    e. Tony:
        i. Thinkpad w/ Linux
        ii. Desktop w/ Windows 10

Revised personas

Persona 1:

Name: Hanz Hanson
Pos / Env: Assistant professor of Astronomy @UTSC
Gender/Age: Male, 40
Skills:
- Familiar with Unix, github, and programming in general.
- Experienced with python, app development, etc.

Personality:
- Detail-oriented, likes to stick to schedule.
- Very methodical, precise, organized
- Likes to keeps track of latest developments in Physics and Astrophysics (studies scientific publications, both print and web-based)
- Skeptical about research information he is presented. Always carefully consults third party sources before trusting any scientific discovery or opinion.

Attitude towards tech:
- Strongly prefers open source and GNU software and systems
- Prefers UNIX over NT

Goals when using our program:
- Wants to maintain most up to date planet catalogue
- Needs our program to simplify and automate the task
- Does not require user privileges because access to the github database acts as an authentication

Persona 2:

Name: David Davidson
Pos / Env: Graduate Student in Astrophysics @ York University
Gender/Age: Male, 30
Skills:
- Familiar with MacOS and Windows only.
- Does not have background in programming or using complex applications.
- Can easily replicate simple procedures when given.
- Likes it when a good User Manual is available
- Has used Command Line Interface before, but is not very comfortable with it.

Personality:
- Somewhat impatient, wants to get work done without complications.
- Busy person, needs ways to optimise his working routine
- Has limited time to spare to work on his side projects (one of them is Open Exoplanet Catalogue)

Attitude towards tech:

- Comfortable with using computers in his daily life
- Comfortable with using any internet browser on MS windows
- Easily confused by complex interface, prefers simplicity

Goal when using our program:

- Not related to the research of Hanz Hanson but is fond of the Open Exoplanet Catalogue and wants to suggest updates to the person maintaining the database

Product Backlog:

Priority Scale:

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

1 Story Point (sp) == 1 Developer Hour

USER STORIES:

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: 3, 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: 3, 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

Release Plan

User story dependencies:
1, 2, 3, 4 -> 5, 8, 9, 10, 12 -> Everything else

User stories 1, 2, 3, 4 are crucial for the operation of our program, all other stories depend on the completion of those. These stories, plus 5, 8, 10, 12 constitute the most basic operation of our program. All other stories constitute desired secondary features, to make the operation easier and to prevent potential inconsistencies in the updates.

Priority of user stories: see product backlog
Estimated time of completion for each user story: See product backlog

Expected Total: 150 Hrs.
Product velocity(approximation): 25 hours per sprint.
(5 hours per person per sprint)

Goals of first release October 17 - October 24:

- Complete user story 1, 2 and 3
- Ensure that even though only stories 1, 2 and 3 will be completed that the program remains working and bug-free.
- Ensure that the program is operable from the command line

Goals of second release October 24 - October 31:

- Complete user story 4, 5, 8 and 9

Goals of third release October 31 - November 7:

- Complete user story 6, 7, 10, 11, 12, 14
-

Goals of fourth release November 7 - November 14:

- Complete user story 15, 16, 17, 18, 23, 24

Goals of fifth release November 14 - November 21:

- Complete user story 13, 25, 26, 27

Goals of sixth (last) release November 21 - November 28 / December 1:

- Complete user story 19, 20, 21, 22

Sprint Backlog

Sprint lengths: 1 week

For the first sprint, we decided to select user stories 1, 2 and 3, which describe the retrieval of the information from different catalogues.
Tasks will be assigned in the following format: UserStoryNumber-TaskNumber

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

    1. Create driver.py, that contains the main method and that can be invoked from the command line and cooperates with all other classes (sp: 1)
    2. Implement option parsing in the main method. (sp: 1)
    3. Implement the “update” mode of operation invoked from the command line (sp: 1)
    4. Create planet class to represent a single planet and store the data for the fields. (sp: 1)
    5. Create a star class to act as a storage of info about one star and its system (including references to planets) (sp: 1)
    6. Create downloader class for Open Exoplanet Catalogue (to retrieve information from OEC into a list of planet objects) (sp: 2)
    7. Unit test downloader class (sp: 0.5)
    8. Test option parsing of the main method. (sp: 0.5)

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. Create downloader class for NASA catalogue to get info in CSV format. (sp: 1)
    2. Create downloader class for exoplanet.eu to get info in CSV format (We need separate downloaders for the 2 catalogues because exoplanet.eu does not provide an API, just a link, that is why downloading databases requires two separate algorithms).(sp: 1)
    3. Create parser for NASA catalogue to convert the CSV into a list of planet objects. (sp: 2)
    4. Create parser for exoplanet.eu to convert the CSV into a list of planet objects. (sp: 2)
    5. Test parser for NASA. (sp: 1)

6. Test parser for exoplanet.eu (sp: 1)

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

    1. Implement “Proposed change” class, which represents a single change between the catalogue (and has links to 2 planet objects, and has additional information about the field that was modified, source catalogue, date) (sp: 3)
    2. Implement “Comparator” class that will process the datasets from Open Exoplanet Catalogue and outputs a list of Proposed Change objects. (sp: 12)
    3. Implement “show changes” mode of operation. (sp: 2)
    4. Test the Comparator class(sp: 3)

Rough Plan of Who Does What and When:

During sprint 1:

(Approximately 5 story points per person per sprint)
Each user story is broken down into multiple tasks.

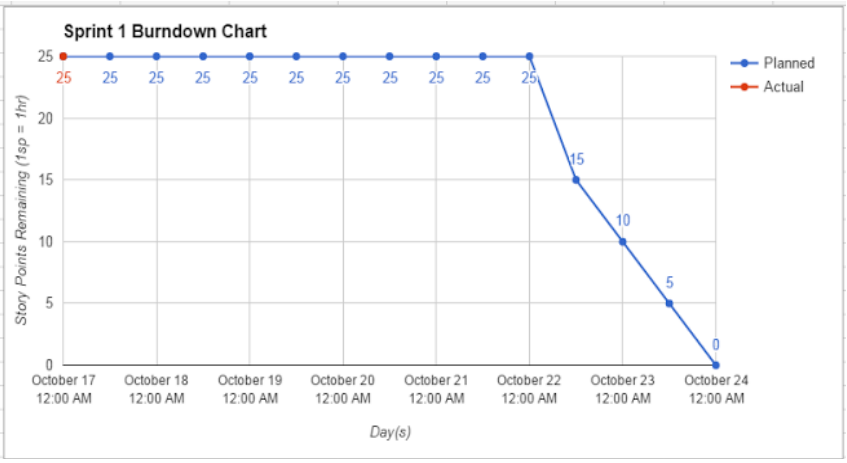
User story 1: Jerry (5 sp) (Tasks: 1, 2, 3, 4, 5) + Eric(3 sp) (Tasks: 6, 7, 8) - sometime before the end of Friday.
User story 2: Albion (5 sp) (Tasks: 1, 2, 3, 5) + Tony(3 sp) (Tasks: 4, 6) - sometime before the end of Friday.
User story 3: Vasili (5 sp) (Tasks: 1, 3) + Eric(2 sp) (Tasks: 2 (in collaboration with Tony)) + Tony(2 sp) (Tasks: 2 (in collaboration with Eric)) before the end of Sunday.

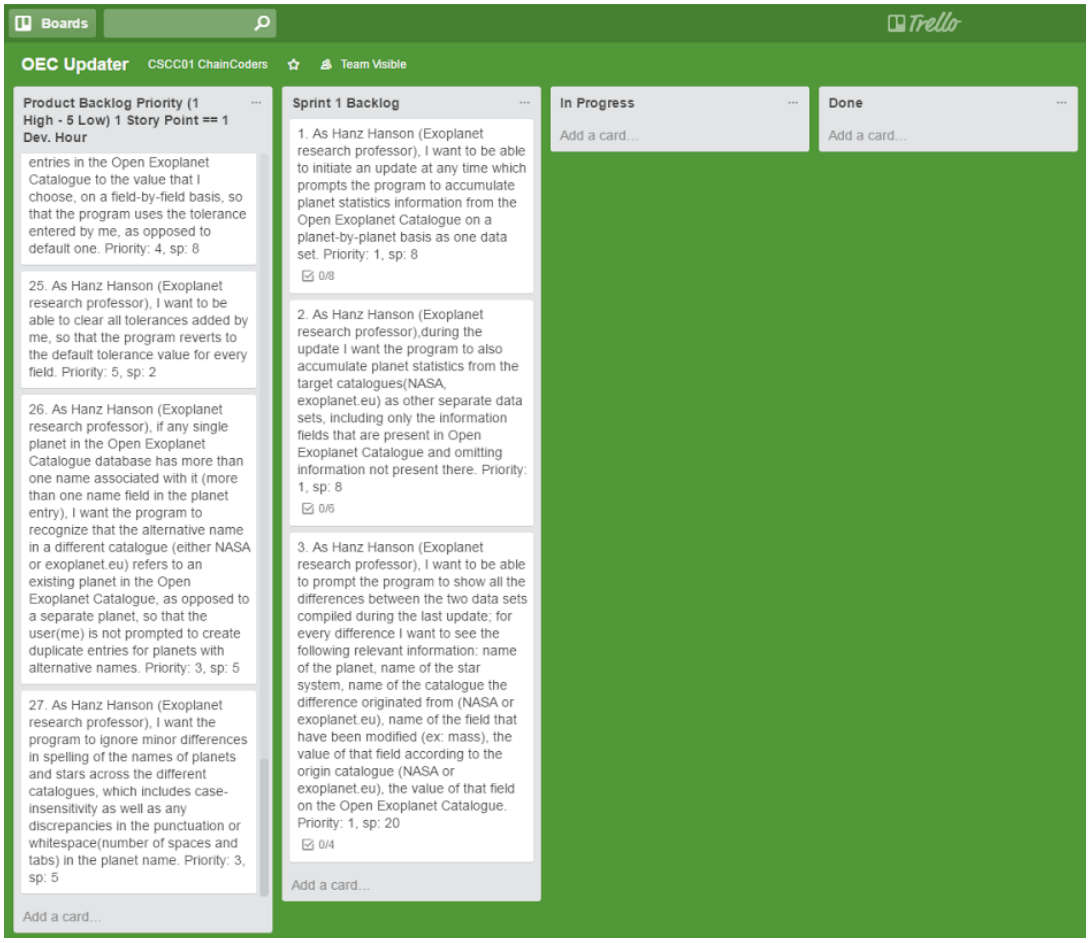
We are going to dedicate hours remaining after completing user stories 1 and 2 to complete as much of user story 3 as possible. Tasks 1, 2 and 3 are the most important parts of it. The deliverable will be in the working state after the end of Sprint 1. The functionality to see a list of proposed changes will be implemented at least partially.

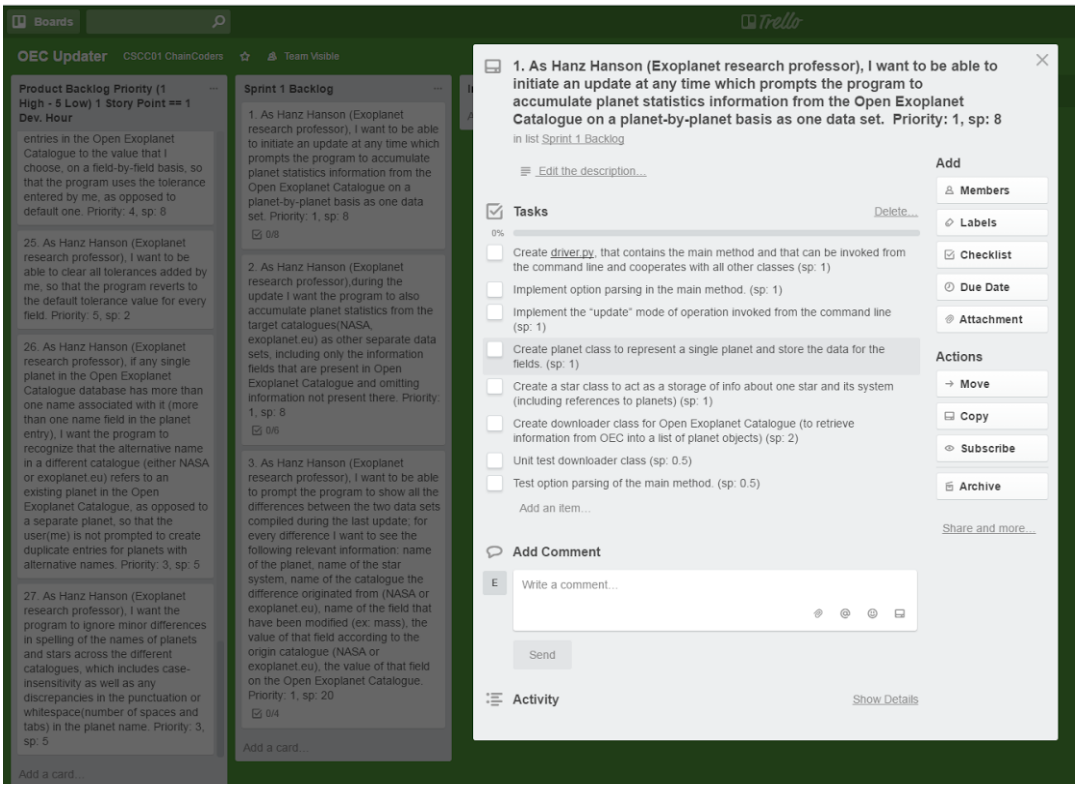
Because of the course work for other classes we can not point out a more precise time of completing the tasks. Team members are required to fully complete their tasks sometime before Sunday afternoon at every sprint. Postponing work until the weekend is acceptable in extraordinary circumstances (midterm tests, large assignments). Scrum meeting takes place in person on Wednesday, and online on Saturday. Vasili is taking minutes.

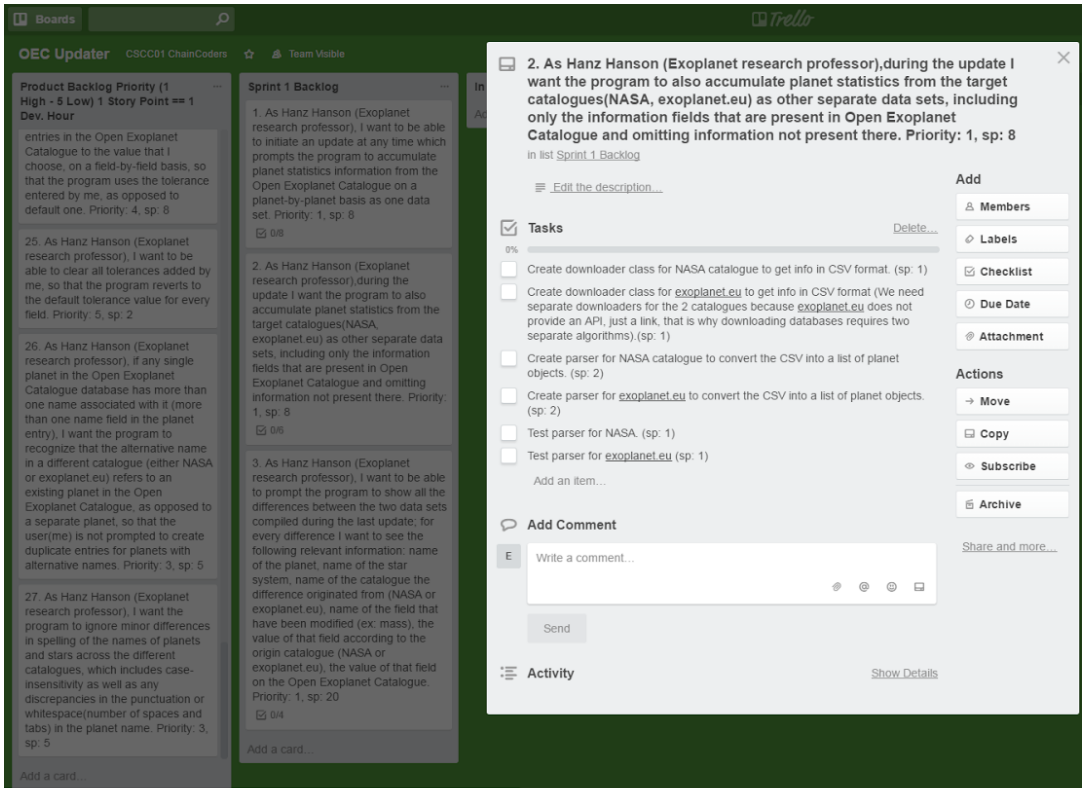
Snapshots of set up burndown chart and set up taskboard

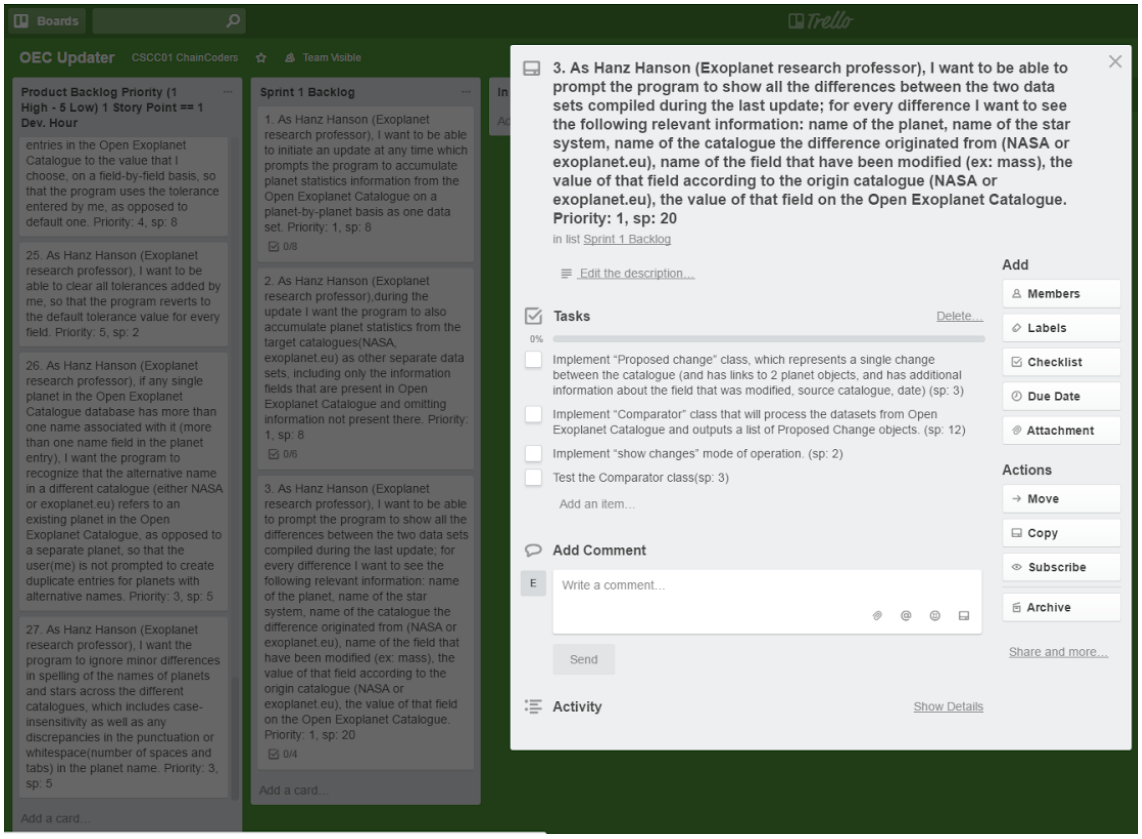
Day(s)	Planned	Actual
October 17 12:00 AM	25	25
October 17 12:00 PM	25	
October 18 12:00 AM	25	
October 18 12:00 PM	25	
October 19 12:00 AM	25	
October 19 12:00 PM	25	
October 20 12:00 AM	25	
October 20 12:00 PM	25	
October 21 12:00 AM	25	
October 21 12:00 PM	25	
October 22 12:00 AM	25	
October 22 12:00 PM	15	
October 23 12:00 AM	10	
October 23 12:00 PM	5	
October 24 12:00 AM	0	





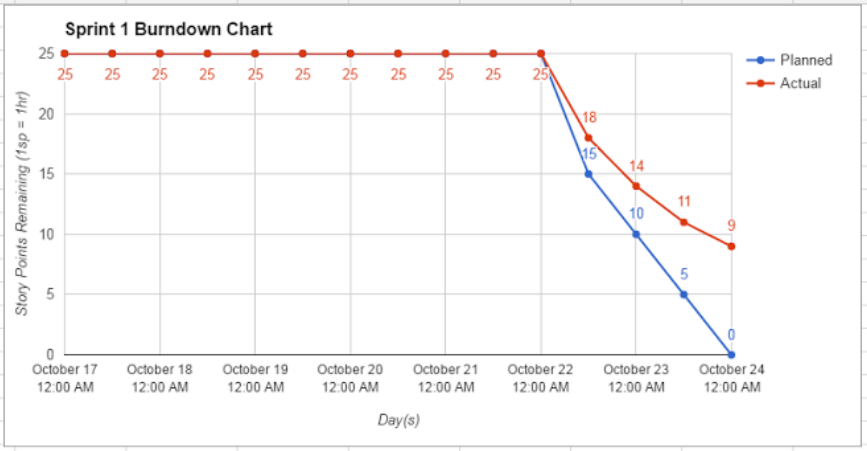

Boards
Trello
OEC Updater
CSCC01 ChainCoders
Team Visible
Product Backlog Priority (1 High - 5 Low) 1 Story Point == 1 Dev. Hour
entries in the Open Exoplanet Catalogue to the value that I choose, on a field-by-field basis, so that the program uses the tolerance entered by me, as opposed to default one. 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
Add a card...
Sprint 1 Backlog
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
0/8
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
0/6
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
0/4
Add a card...
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
in list Sprint 1 Backlog
Edit the description...
Tasks
Delete...
0%
Create driver.py, that contains the main method and that can be invoked from the command line and cooperates with all other classes (sp: 1)
Implement option parsing in the main method. (sp: 1)
Implement the "update" mode of operation invoked from the command line (sp: 1)
Create planet class to represent a single planet and store the data for the fields. (sp: 1)
Create a star class to act as a storage of info about one star and its system (including references to planets) (sp: 1)
Create downloader class for Open Exoplanet Catalogue (to retrieve information from OEC into a list of planet objects) (sp: 2)
Unit test downloader class (sp: 0.5)
Test option parsing of the main method. (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...

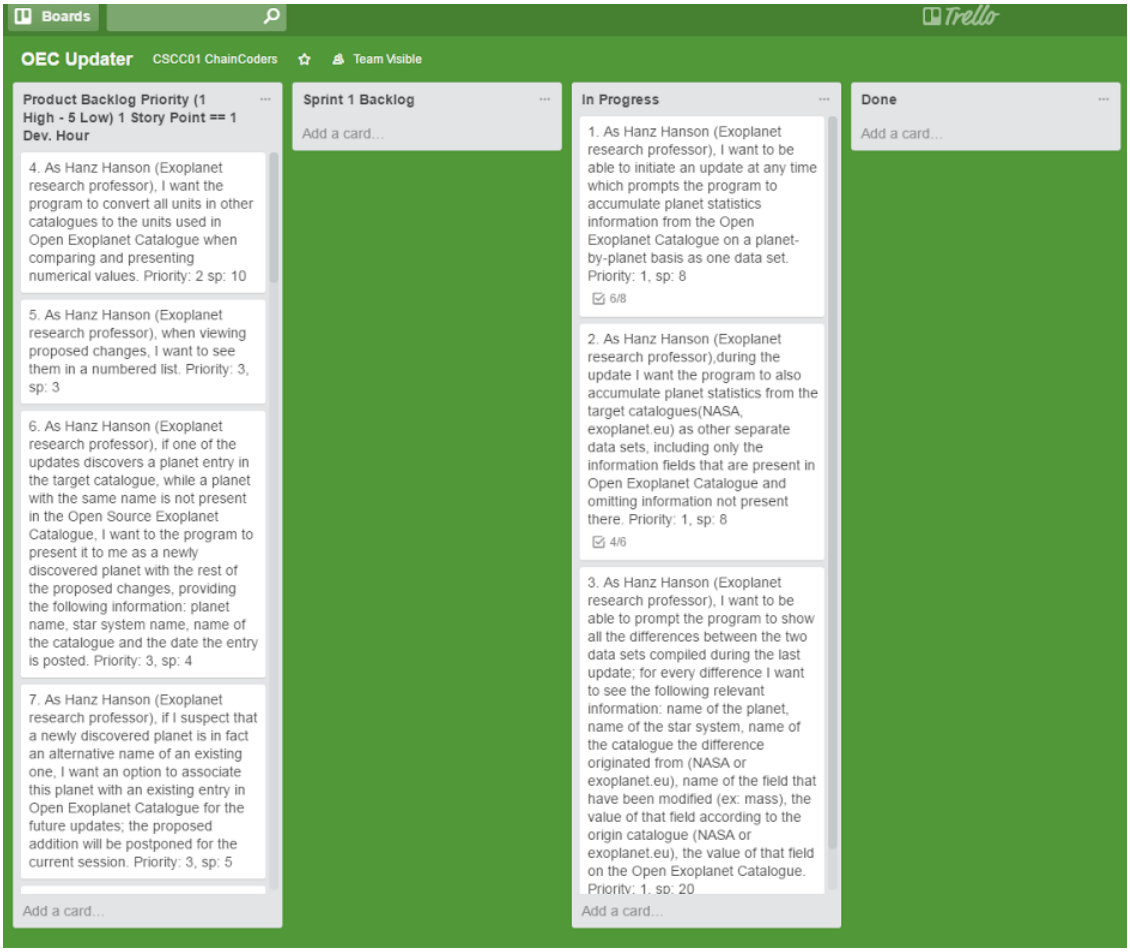

Boards
Trello
OEC Updater
CSCC01 ChainCoders
Team Visible
Product Backlog Priority (1 High - 5 Low) 1 Story Point == 1 Dev. Hour
entries in the Open Exoplanet Catalogue to the value that I choose, on a field-by-field basis, so that the program uses the tolerance entered by me, as opposed to default one. 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
Add a card...
Sprint 1 Backlog
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
0/8
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
0/6
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
0/4
Add a card...
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
in list Sprint 1 Backlog
Edit the description...
Tasks
Delete...
0%
Create downloader class for NASA catalogue to get info in CSV format. (sp: 1)
Create downloader class for exoplanet.eu to get info in CSV format (We need separate downloaders for the 2 catalogues because exoplanet.eu does not provide an API, just a link, that is why downloading databases requires two separate algorithms).(sp: 1)
Create parser for NASA catalogue to convert the CSV into a list of planet objects. (sp: 2)
Create parser for exoplanet.eu to convert the CSV into a list of planet objects. (sp: 2)
Test parser for NASA. (sp: 1)
Test parser for exoplanet.eu (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...


Boards
Trello
OEC Updater
CSCC01 ChainCoders
Team Visible
Product Backlog Priority (1 High - 5 Low) 1 Story Point == 1 Dev. Hour
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
Add a card...
Sprint 1 Backlog
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
0/8
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
0/6
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
0/4
Add a card...
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
in list Sprint 1 Backlog
Edit the description...
Tasks
Delete...
0%
Implement "Proposed change" class, which represents a single change between the catalogue (and has links to 2 planet objects, and has additional information about the field that was modified, source catalogue, date) (sp: 3)
Implement "Comparator" class that will process the datasets from Open Exoplanet Catalogue and outputs a list of Proposed Change objects. (sp: 12)
Implement "show changes" mode of operation. (sp: 2)
Test the Comparator class(sp: 3)
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 of Sprint Burndown Chart and Taskboard

	Day(s)	Planned	Actual
2	October 17 12:00 AM	25	25
3	October 17 12:00 PM	25	25
4	October 18 12:00 AM	25	25
5	October 18 12:00 PM	25	25
6	October 19 12:00 AM	25	25
7	October 19 12:00 PM	25	25
8	October 20 12:00 AM	25	25
9	October 20 12:00 PM	25	25
10	October 21 12:00 AM	25	25
11	October 21 12:00 PM	25	25
12	October 22 12:00 AM	25	25
13	October 22 12:00 PM	15	18
14	October 23 12:00 AM	10	14
15	October 23 12:00 PM	5	11
16	October 24 12:00 AM	0	9





Brief overview of the project's progress:

During our first spring we were able to make substantial progress in information retrieval and parsing, however we encountered some unexpected problems, most notably with "requests" library, which is not in Python's standard library, but is required in order to retrieve CSV data from NASA archive. Using "requests" caused us to experience setbacks and do extra work in order to ensure compatibility with different environments. We ended the sprint behind, however we gained valuable understanding of how the project is supposed to function, and we are prepared for the next sprint.

Estimated project velocity was: 25 sp / sprint.
Actual project velocity: 16 sp / sprint.

We tried to follow our original plan as close as possible, however we soon discovered some unforeseen challenges with data processing, network libraries and compatibility. While many of us put in more hours of work than we planned, we found the project progressing slower than expected. The lead reasons would be our lack of experience building applications that interact with web interfaces. We had to change our plans in the last two days, pushing User story 3 (data comparison) part into the next sprint. What we have now is a working application that can retrieve and parse data from Open Exoplanet Catalogue and exoplanet.eu. We are preparing for the next sprint and we are sure we can plan better to avoid further setbacks.