

Sprint 1 Report

Plannertarium, by Team Scrumbags, 10/24/2023

Actions to stop doing

The team should stop being late to scheduled events because the rest of the team must wait until the missing member shows up or doesn't show up. The team should stop rushing through scrum meetings because we do not usually approach the 15 minute mark, and our swiftness is coming at the cost of being less informative. The team should stop procrastinating project work because we are falling behind schedule, especially with our new sprint plan.

Actions to start doing

The team should use our experiences in sprint 1 to build a more realistic plan for sprint 2 because we drastically underestimated the amount of time it took to learn. The team should clearly split up class logistical work because it often gets put off to the last minute. The team should put more time into the project because we were not able to get very far this sprint.

Actions to keep doing

Standup meetings are working well because it gets the team synced and tasks set. Source control is working well because we have not had issues with conflicting files in the repository yet. Frequent online communication is working well because we can answer each other's questions quickly while we are working on the project and avoid delays.

Work Completed/Not completed

Completed tasks are crossed out. We did not finish any user stories.

As a student, I want to be able to...

1. see my inputted information when I reopen the app so that I do not need to keep it open
 - a. ~~Create a Firestore Database - 1h~~
 - b. ~~Add proper rules to the database - 1h~~
 - c. Add user authentication to track user accounts - 2h
 - d. ~~Create a data format for tasks in the database - 1h~~
 - e. ~~Create a class in Dart to represent tasks - 1h~~
 - f. ~~Create a data format for events in the database - 1h~~
 - g. ~~Create a class in Dart to represent events - 1h~~

- h. Create basic unit tests for the database - 2h
 - i. Create methods to read user data from the database into memory upon app launch - 1h
- 2. create tasks so that I can see what my tasks are - 9h
 - a. Create a new Widget on the UI to accept new tasks - 2h
 - b. Display tasks in daily view - 2h
 - c. Display tasks in weekly view - 2h
 - d. Display tasks in monthly view - 2h
 - ~~e. Create methods that save a task object to the database - 1h~~
 - f. Link the UI Widget to the new method - 0h
- 3. change properties of a task so that I can mark them as completed, move to another date, etc - 7h
 - a. Add UI for swipe left on a task Widget - 1h
 - b. Add UI for swipe right on a task Widget - 1h
 - c. Add UI for double tap on a task Widget - 1h
 - d. Add UI for holding down on a task Widget - 1h
 - e. Add UI for moving tasks to different times on the timeline - 2h
 - f. Add getters and setters to Task class to update their attributes - 1h
- 4. create events so that I can see what my events are - 9h
 - a. Create a new Widget on the UI to create new events - 2h
 - b. Display events in daily view - 2h
 - c. Display events in weekly view - 2h
 - d. Display events in monthly view - 2h
 - ~~e. Create methods that save an event object to the database - 1h~~
 - f. Link the UI Widget to the new method - 0h
- 5. change properties of an event so that I can give them details about location/time/etc - 4h
 - a. Add UI for moving events to different times on the timeline - 3h
 - b. Add getters and setters to Event class to update their attributes - 1h
- 6. switch between time windows so that I can plan for the short, medium and long term as needed - 7h
 - a. build the screen for the daily view - 1h
 - b. build the screen for the weekly view - 1h
 - c. build the screen for the monthly view - 1h
 - d. add UI buttons and swipes at the top - 2h
 - e. implement buttons that change the current view - 2h
- 7. view the current event so I can quickly see and take note of what is happening - 7h
 - a. Add check to see if the current time overlaps with an event - 0h
 - b. Create a new UI Widget to indicate an event that is currently happening - 2h
 - c. Link the check to the Widget - 0h
 - d. Add a progress bar to the new widget - 2h

- e. Implement a button making the widget take you to the event view - 1h
- f. Add a progress bar to the event view UI - 2h

Work Completion Rate

Total number of user of stories completed during the prior sprint: 0

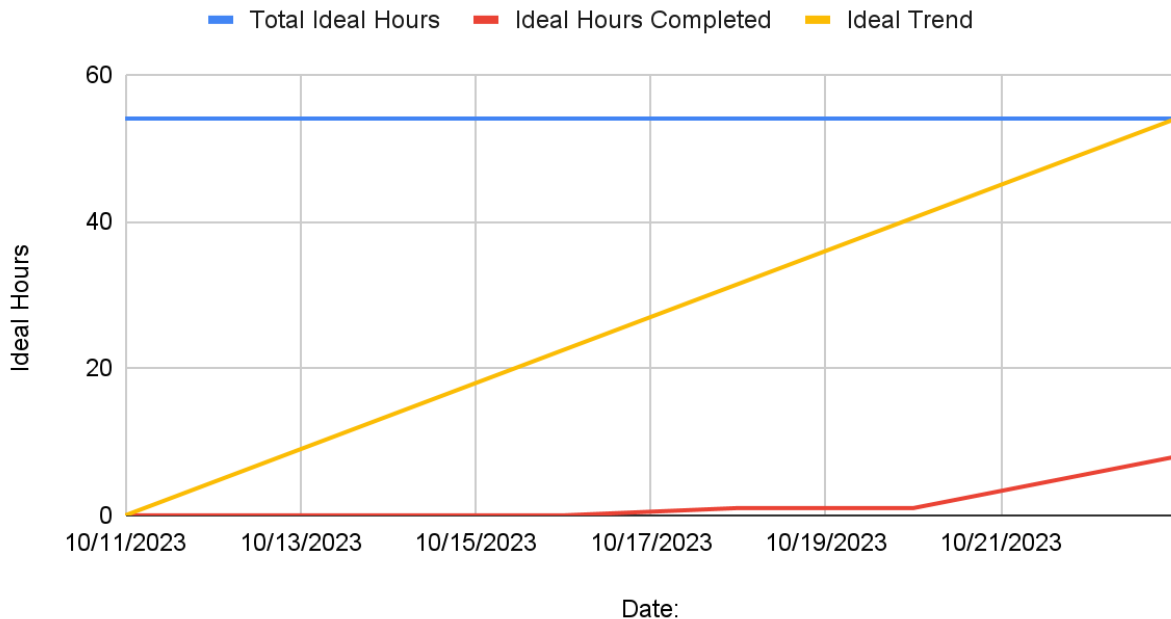
Total number of ideal work hours completed during the prior sprint: 8 hours

Total number of days during the prior sprint: 14 days

Stories/day = $0/14 = 0$ stories/day

Ideal hours/day = $8/14 = 0.571$ ideal hours/day

Sprint 1 Burnup Chart, Plannertarium



We were unable to get more work completed for a number of reasons. Firstly, our initial estimation for total ideal hours for all our user stories this sprint was about 50 hours. However, none of us really knew much about how long these tasks would take due to a lack of experience among us. Then, after overcoming some spikes and watching tutorials on the language, we re-evaluated our tasks and came up with a lower total ideal hour estimate of around 27. Then, after our first TA meeting on 10/16/2023, we were told to break up/redo all our user stories and tasks, to which we assigned new estimates with a little bit more knowledge. This total was 54 ideal hours. However, the spikes were very large, and we each spent much less time working on the development than expected at first. Even though we spent the last few days working very hard and long hours, we were only able to achieve 8 ideal hours of work, less than one fourth of the actual hours we spent working on development as a team. We also spent a lot of time on the documents for scrum, since we were new to it and were still a little unsure about it. We enter into

the next sprint with a better understanding of what user stories are and some spikes completed, so will plan accordingly.