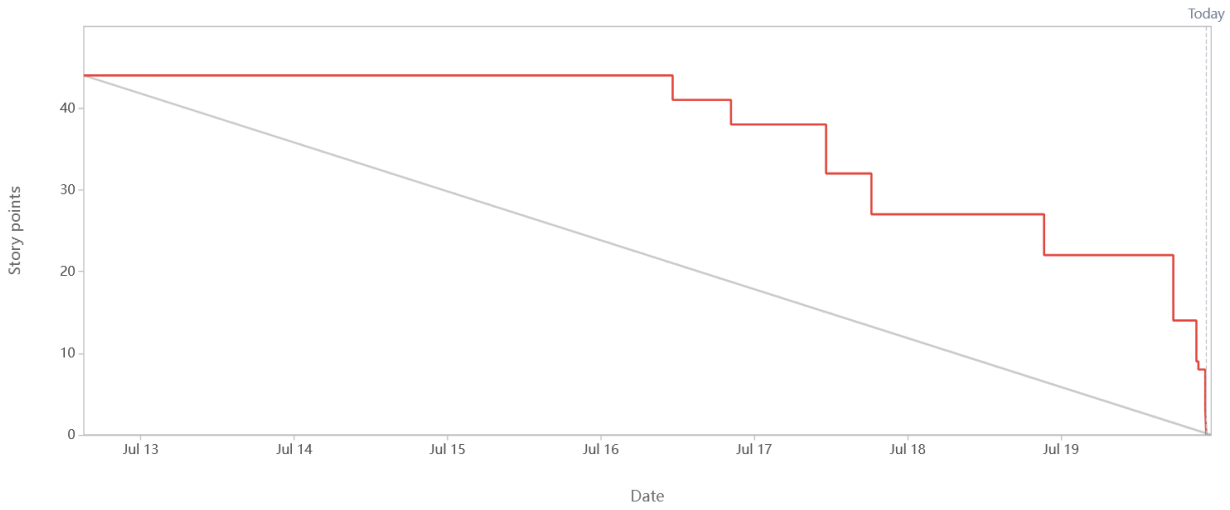
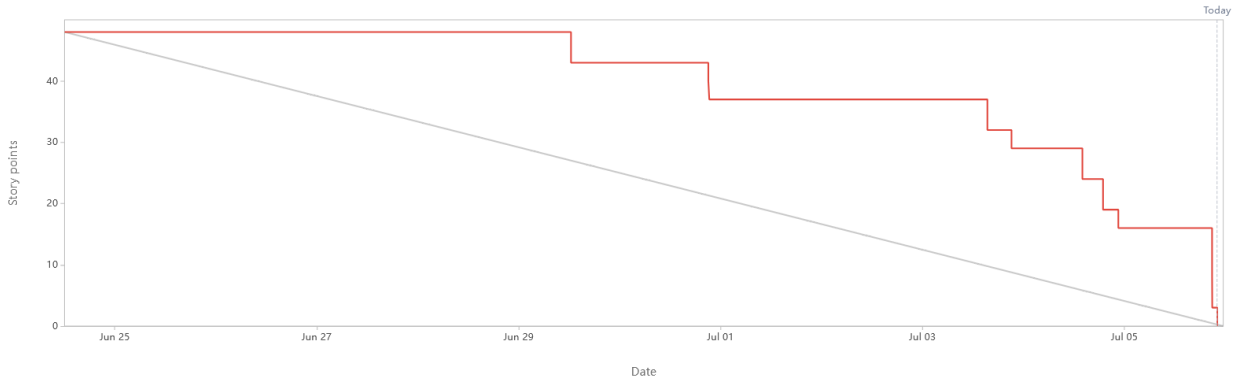
Sprint 3 - Burndown Chart

Sprint 3



Sprint 2



Analysis

Based on story points, our actual burndown curve differed greatly from the guideline and showed high similarity to that of the last sprint. During this sprint, the first story points began to be completed on July 16th, which is more than halfway through the sprint. This pattern repeats what was seen during Sprint 2. This suggests that our attempts to address the issues of Sprint 2 were not very effective:

- Smaller tasks were only partially successful at addressing low story point completion at the start of the sprint. While we created more stories with story point estimates of 1 or 3, instead of 5, this did not seem to have an effect on when the first stories were completed. While stories did allow rapid progress to occur when the first pull requests were made, it did not affect when the first pull request was created.
- It is evident that we will have to discuss as a group to agree on how much we value having a smoother burndown curve. Any change to our future burndown curve will likely

have to involve us making meaningful changes to our schedules and personal priorities, in order to tackle stories sooner.

The delay in creating the pull requests was worse than the past sprint. However, we were able to complete story points more quickly once members started pushing code and creating pull requests. This is visible in the burndown chart by the steeper slope of the burndown chart from July 16th to the sprint end date. This was due to two reasons:

- We put more consideration into selecting user stories from the backlog when preparing the sprint. We prioritized stories that were already related in some way to previous stories, so that members were already familiar with some of the story requirements. For example, refactoring of the map tracking interface was handled by the member who last modified it.
- We made a conscious effort to assign stories to minimize blockers and accommodate technical expertise in our team. We divided larger tasks up to minimize the possibility of members blocking one another, such as when they modified identical user interface elements. When we first assigned tasks, we also discussed as a group whether there were any foreseeable blockers for each person, so that everyone was aware ahead of time whether they should anticipate them.

Velocity

This sprint, we started with 44 story points which is slightly lower than 48 from the previous sprint. So velocity decreased slightly, even though we assigned the same number of tickets (12). The lower story point number is partly because we created and tackled more "clean-up" tickets during this sprint, which involved code refactoring and patching of edge cases. While each clean-up ticket had fewer story points than the typical user story ticket, it still took considerable time to get these tickets properly reviewed and merged into the develop branch.

We foresee that our next sprint may have slightly fewer story points at the beginning, since we have thus far completed most major features. Hence, future tickets will mostly be visual UI improvements, bug fixing, and final touches to the authentication and authorization system, each of which should be smaller than our traditional user stories. More time must be dedicated during sprint 4 for testing and identifying important refactoring tasks.