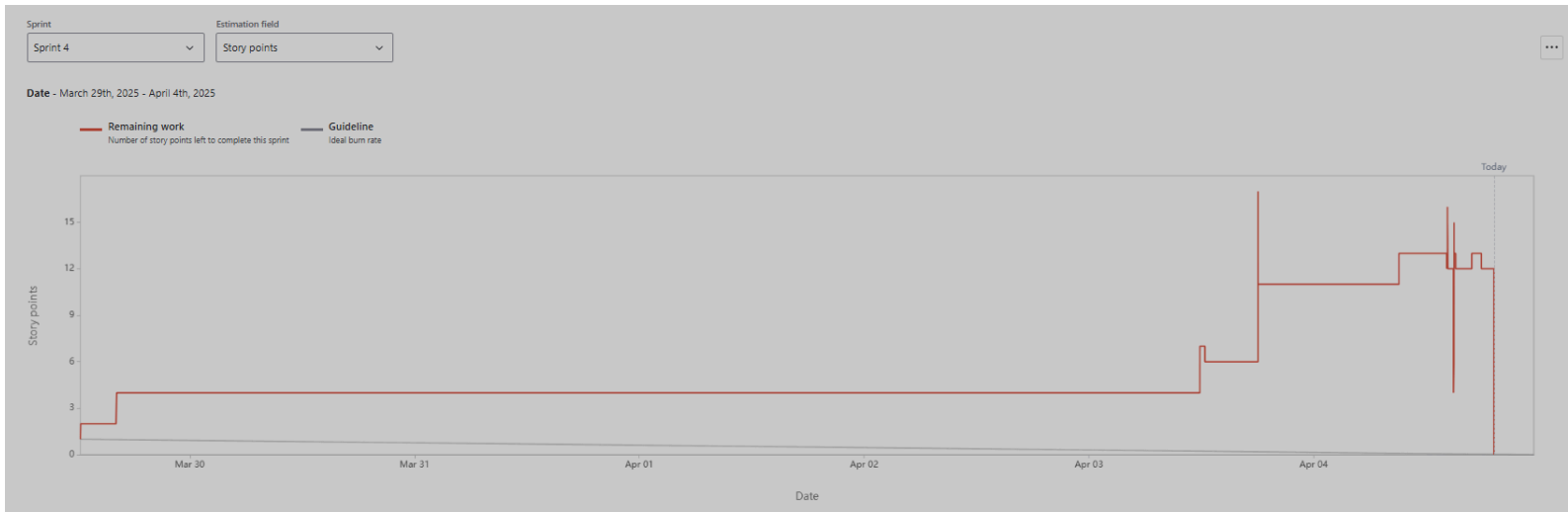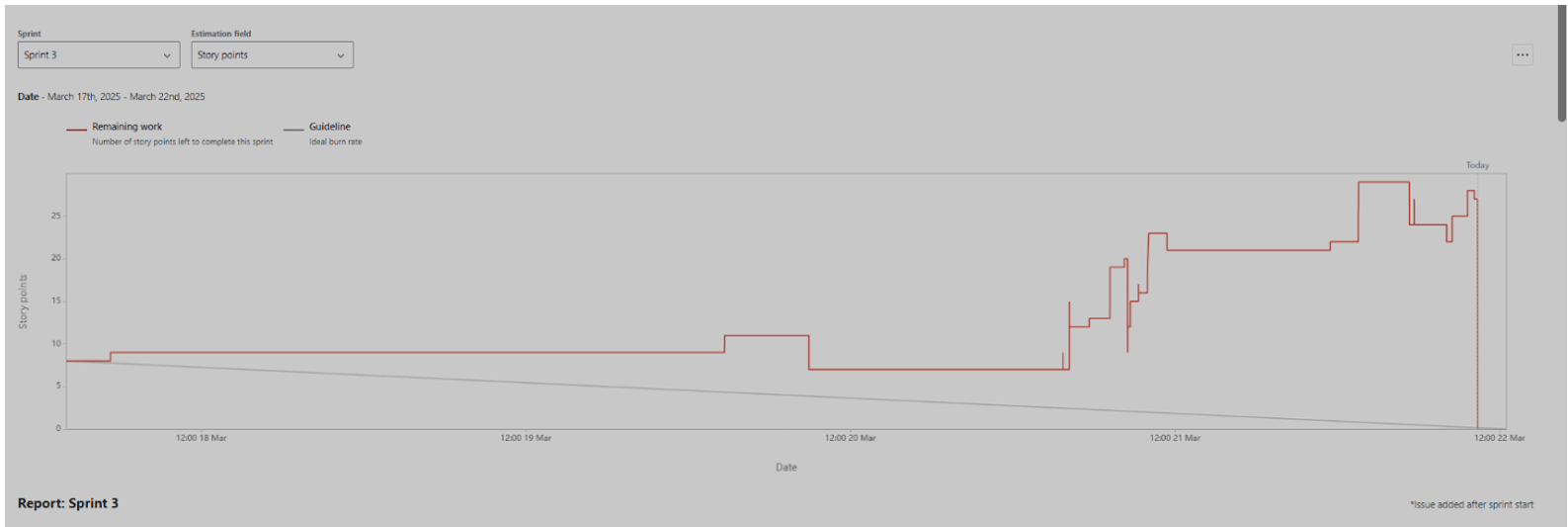Sprint 4 chart:

Sprint 3 chart:

Comments/Analysis:

Looking at the burndown charts for the two sprints, we can see that both sprints experienced tasks being added throughout the duration of the sprint. This likely came from the fact that we originally created stories without splitting them into Core and Client components since we just wanted to see what work needs to be done in general. However, while developers were working through the sprint and taking on tickets, they would split up the tickets into the frontend and backend components for clarity and to accurately divide responsibilities. In Sprint 3, we also see that a lot more tasks were being added than in Sprint 4, and this is because we wanted to get as many features as possible into Sprint 3 before the final in-class demo. In Sprint 4, we were mostly focussed on fixing bugs and finishing leftover pending features. Also, in Sprint 3, we can see that our work was spread out more evenly throughout the sprint. In contrast, Sprint 4 saw most of the work being completed toward the end since all team members were pretty busy with other courses at the beginning of the sprint. Despite this, we were able to complete all of our tasks, since there were significantly less things leftover for us to do in Sprint 4 than Sprint 3. In Sprint 3, we wanted to finish as much as we could before the scheduled demo during our

lecture. The team was more motivated to finish as much as possible so we would have more functionality to show during our presentation. In Sprint 4, we were just finishing up the pending features and fixing any bugs that came up.

In terms of velocity, during Sprint 3 we completed 59 story points, and during Sprint 4, we completed 35 story points. This decrease for Sprint 4 can be explained by the same reason as mentioned above, and it's the fact that we tried extra hard to get more work in during Sprint 3 so that we had more things to show others during our in-class demo. In Sprint 4, we didn't have as many things left over to do. We planned to do this from the very start (Sprint 0), because we wanted to leave the last sprint to mostly finish up any unfinished features and focus on cleaning up any bugs/issues.