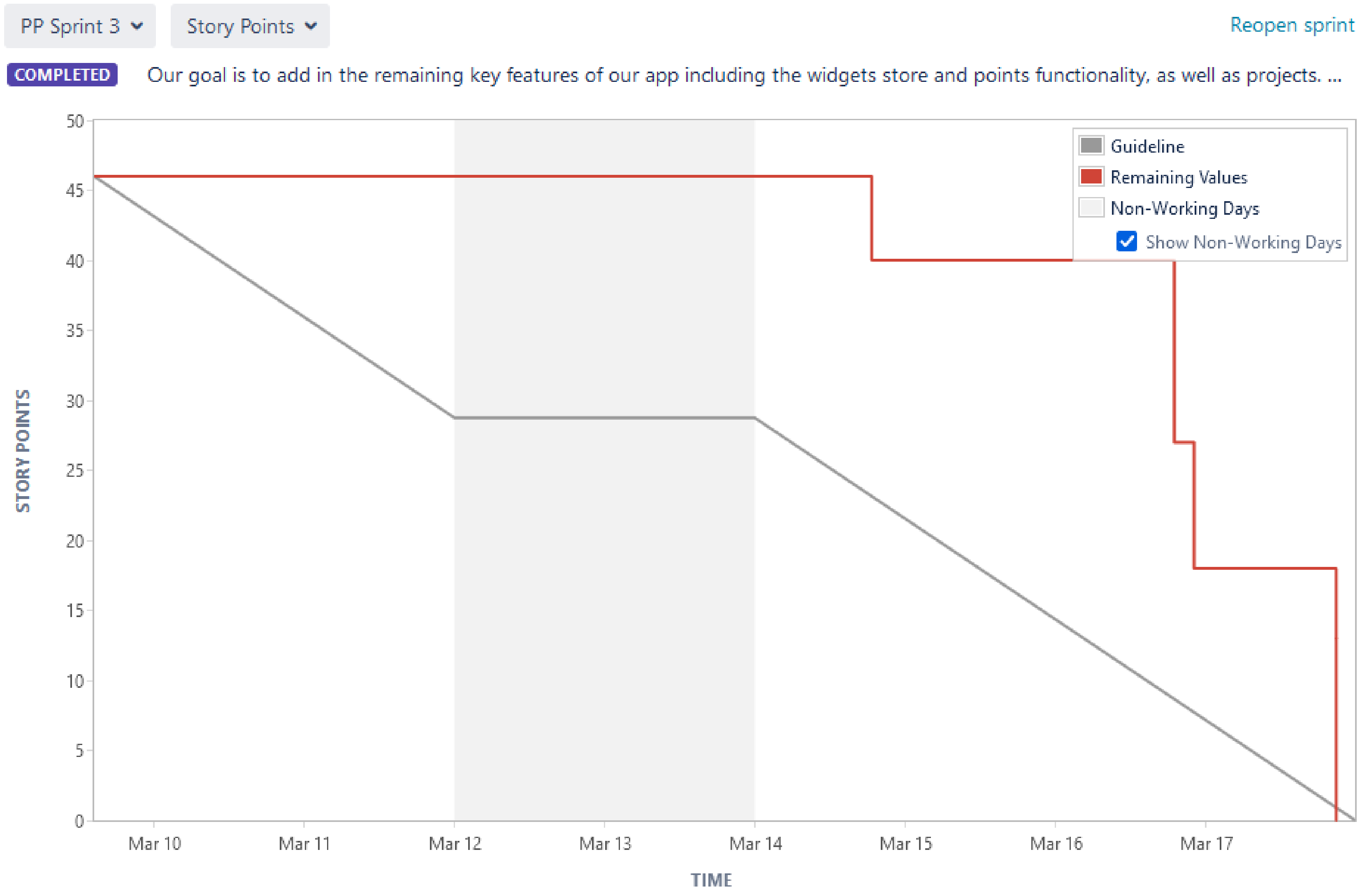
Sprint 3 Burndown Chart



PP Sprint 3 ⌄ Story Points ⌄

Reopen sprint

COMPLETED Our goal is to add in the remaining key features of our app including the widgets store and points functionality, as well as projects. ...

Introduction

Our Sprint 3 velocity was 46 points, down 30 points from 76 in Sprint 2. This is because this time we took on less user stories due to having to accommodate a delayed start (we lost 4 days of work because we thought we weren't allowed to push to repo until we got our marks back, like previous sprints). Similar to Sprint 2, we also had two Developer Stories/"chores" this time. These were crucial in refining the existing code base and setting the stage for future long-term development. Perhaps the most important was PP-68, which refactored class components into functional, so every feature was similarly designed and communicated via props. This user story also amalgamated component styling from a mix of inline, .CSS, and .style.js files into one medium. These changes make the code base less complex to read and understand. Our other 3 stories for a total of 5 on Jira were demo-able user features: point accumulation and display, widget store to purchase and add widgets, and finally linking Projects with tasks.

Results

Our velocity changed significantly from last sprint because we didn't have *as essential* developer stories this time around. Although we still did want to refine the codebase as we have done, we weren't forced to take on excessive developer stories like last time in addition to the three demo-able features we needed. The scope of the work in conjunction with our other courses culminating towards the end of March fit reasonably. Our sprint average still stayed within recommended lecture estimates, around 40-50 points across the 3 sprints so far.

Looking at the burndown chart, we were able to stagger the user stories completion a little better this time around. PP-9 was able to be completed on the Monday, although smaller later revisions were needed once other members merged their code in. We noticed this was a recurring theme in Sprint 3. PP-4 and PP-10 were completed on March 16, and PP-68 and PP-69 were completed on March 17. This was an improvement from last sprint as everything was completed in the final 1-2 days. This time we used the whole week.

We hope for Sprint 4 to get an appropriate start (confirmed date via S0 handout this time with the team!). A good result of Sprint 3 however was having a more manageable workload, which ultimately meant not having to push user stories to next sprint like we were forced to for Sprint 2.

Lessons Learned

Perhaps the most important take-away from Sprint 3 was the value and impact a refactoring developer story like PP-68 can cause. As it was changing the nature of the components themselves, it would have been *much* better and cleaner to let the team member complete that at the start of the sprint, then everyone pull from development and build off the renewed codebase. Instead, our team coded additional functionality, features, and styling even as PP-68 work was being done. This led to massive merge conflicts, refactoring rewrites, and required pushing additional commits to user stories that were already marked done previously. This headache could have been avoided through better Sprint planning, and is definitely a decision factor we plan on taking into Sprint 4. This was an extremely important lesson in the software development process that we are thankful for having learnt in an education setting rather than the workplace, where real-world impacts would have occurred, and stress would be significantly greater.