University of Kent

CO553 Sprint One Report

A report consisting of our sprint review and retrospective

Group C2-3

Agile Development and Software Security B (CO553)

CO553 Scrum Project Report

The aim of our project is to create a dynamic cinema website that can be used to book tickets for films. The website will retrieve and display data from a collection of relations we have in our database. Additionally, it uses user input to write to some of the relation (e.g. register and time/ticket selection page).

Instead of assigning the traditional roles of product owner, scrum master, team member 1 and team member 2, we decided to divide our team into a scrum division and a coding division. Two members would handle the creation of product and sprint backlogs after brainstorming with the team, while the other two members would work on the code for the website.

We believed that by working in pairs, members can bounce ideas of one another and assist each other with issues. Although we made sure to be flexible with our work and assist the other division if they found something challenging.

Creating the Product and Sprint Backlog

In our first scrum meeting we established that the product and backlog were a priority and so decided to start work on that immediately. In order to do this, we brainstormed several user stories, we set a target of at least 20 user stories. We made sure to break down our epic stories into multiple smaller user stories.

For our second sprint we again assigned points to each user story in terms of importance and how long we wanted to spend on each task, once the points were produced, as a group we assigned each other multiple tasks to complete for the second sprint.

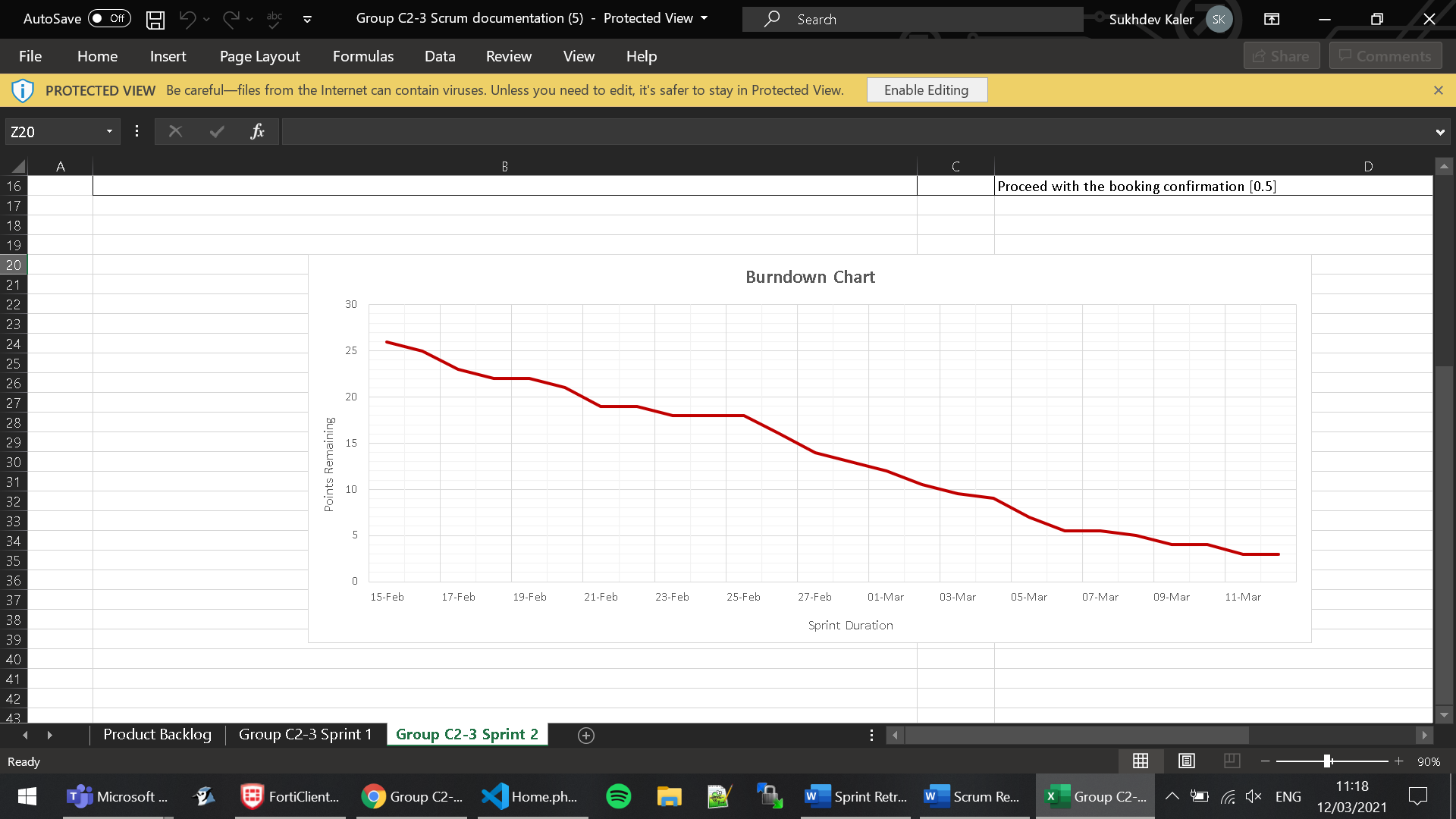


Figure 1: Product backlog

After completing our product backlog, we entered into a discussion about our team velocity. Taking into consideration our current workload and the possibility of it increasing due to other assignments, we decided on a range between 23-25 points per sprint. We then took a collection of user stories who’s point equated to 23, which happened be the first 6 user stories in our product backlog and moved them into our first sprint backlog.

Figure 2: Sprint backlog 3

Figure 3: Burnout diagram



In the sprint backlog, we broke each user story into smaller manageable tasks and assigned points to each task that would total to the point score of its corresponding story. These tasks were spread throughout the 19-day sprint. This was documented on a spreadsheet we made that was inspired by the template on Moodle. The table was designed to display the remaining points after each day. These values were then used to plot a burnout diagram.

Sprint Review

Upon completion of our first sprint as a group, we had a meeting to discuss the remaining tasks of our assignment. Since our first sprint, we have been meeting up more frequently and communicating more effectively. For everyone to keep in close contact, we made a group chat on Discord which is now used as the main platform for sharing specific information or date changes to our meetings. We achieved most of our goals including getting all our tasks done. There were some challenges that we faced during the sprint which included some minor coding errors but those were resolved quickly.

During our second sprint, we came across some issues with our SQL tables. Upon inspection we found that we have a pair of tables that had very similar information, thus making one of them redundant. As such, we attempted to remove the table formerly known as ‘Ticket’, however, we found that the drop operation would not remove the table for us. After conducting researching on this matter and discussing it with one of our lecturers, we discovered that we had to remove the foreign key constraints that were present in the other tables, by doing so we were able to drop the redundant table. From this we learnt that it would be better to construct the tables as we go, that way we have a clear understanding of the contents that need to be within the table, how role the table plays, and it allows us to avoid any other redundancies. These were the main changes that were made during our second sprint.

Overall, our second sprint was very successful and the tasks that was put in our second sprint were completed any remaining tasks as well that were not able to be completed will now be moved onto our third and final sprint to help finish our webpage.

Sprint Retrospective

The communication aspect of our group for

this sprint has been much stronger than in the previous one. We have been setting up regular meetings to discuss our current state of progression and helping each other if a group member needs assistance for a task. The difference between communication at the beginning of the project and communication in sprint 2 is that now we are much more open with each other when it comes to asking for help on a task whereas, in the previous sprint, we tried to fix the issues we faced by ourselves.

Furthermore, one technique that we started using more often in this sprint was pair programming when completing the programming tasks that we assigned to ourselves in our scrum documentation. This method enabled us to be more time-efficient when working on tasks. Moreover, it allowed us to gain insight into different methods that could be used when coding a specific part of our website.

We also began to update the burndown chart as we are doing the code instead of pre-planning it. This way the burndown chart obtained a more realistic representation by showing the period spent on each task.

Finally, we started to use a method called git branching allowing us to create a branch for a specific task. We split these tasks so that two members of the group would work together on a branch and then merge the changes back into the main branch.