Scrum as a part of agile consists of three main roles, Product Owner, Scrum Master, The Team. In our case we as a project group were covering all three roles.

Product owner

Since this was a school project, we naturally didn’t have a product owner. To fill this void in the arguably most important position in the whole SCRUM methodology we all as a group stepped up and came up with ideas in which direction should the product be heading. We all went through the idea generation process, estimation and prioritization therefore we all felt like this was our project. We managed to replicate his usual roles such as creating the product backlog, setting prioritization if the user stories. But some of his roles were not applicable since we owned the product backlog and the release dates were set.

Scrum Master

Coming into the first sprint our idea was to switch the Scrum Master position for every respective sprint. After finishing the first sprint we felt that the position of a Scrum Master was redundant in our workflow, so from that point it was more about who stepped up in the middle of a specific task. During the Unified Process that we tackled with last semester we did not have one assigned Team Leader who would call the shots, there would be someone occasionally stepping up to the leader role but majority of the decision were made democratically, this was the way that we already had experience with for over a year and worked best for us so it came in naturally.

Looking at it retrospectively we found ourselves in situations where having the Scrum Master position and utilizing all of his powers would definitely help us. One of the mentioned situations would definitely be management of the task board during the sprints.

During two sprints we ended up with tasks that were not finished so usually in a situation like this the Scrum Master would intervene and manage the tasks accordingly.

Another example would be that discussion about certain topics took significantly longer because there was not that one person who would pull the hand brake and say “enough, we do it this way”. Our goal during the project was to create the “best” architecture that we possible could. We ended up with refactoring the whole architecture two times during the project. This definitely could have been avoided if a Scrum Master would decide which architecture to use and the time saved would be allocated to another important tasks.

* + - When was he not needed and actually just standing in our way

!!!!LESSON LEARNED KIDS, HAVE YOUR SCRUM MASTER AT ALL COST!!!

Team

Coming into this project we were pretty comfortable with our team composition. Our team as a whole has been working together for over a year so we are used to each other. We know our strengths and weaknesses, some members excel at frontend design others at backend programming, so based on that we would assign roles in the project. But we also like to learn and grow in different fields so that’s why we think pair programming which we used heavily during this project, enabled us to develop the team members in areas they were not so confident in beforehand. Our biggest strength as a team is that we trust each other and we are confident in our individual members decision. We are self organized so we always decided how and when we would work.

With our development methodology based on SCRUM we naturally included the 4 ceremonies which can be also called meetings in our development process.

Sprint Planning

Our usual sprint planning started every Tuesday around 9AM. We would open the Product Backlog, look at the User Stories with the highest priorities and discuss the possible combinations and outcomes. After choosing the highest priority user story we wanted to keep it cohesive so the question always was what other user stories would make a good combination with it. We would also look at our velocity and every time our aim was to target values around 30 velocity. After choosing all the user stories for the sprint backlog we would look at them again as a whole and discuss any doubts. In the next phase we would take the individual user stories one by one and create Task for them accordingly. If we felt a task was too complex we would divided it into smaller tasks, until preferred atomicity. When assigning value of difficulty to tasks, we would often choose one task which was in the “middle” and based on that information we would assign effort values to other tasks. We used the Fibonacci number scale for representing the numbers of difficulty. At the end when everything was set and prepared we would start the sprint. We would commonly refer to the sprints by the name of the most prioritized user story, for example “WPF Sprint” or “Authentication/Authorization Sprint”.

Sprint Review

In the sprint reviews we would always present our latest work from the previous sprint. The reviews were very beneficiary to our group project, it helped us to get the much needed feedback from the teachers. Besides the teachers we got also asked questioned by our fellow classmates. We were also on the other side of the table where we would ask questions. For example we were unsure of the row version implementation but we discussed it with one of the groups and later with teachers. That gave us an idea and assured us that we were on the right track. Getting all this feedback made us realize some possible improvements and helped us to developed a overall better product

Daily Scrums

Daily scrums were a huge part in our development process. In our case we could call it a hybrid between daily scrum and daily stand-up, because we used some practices from XP as well. We would meet almost every day around 9AM. The main questions were: “What did I do yesterday?”, “What will I do today?” and “Do I see any impediment that prevents me or the development team from meeting the sprint goal?”. We would take random turns until every group member would answer these questions. After that we would inspect the sprint task table and use these three questions as a guide to see how we are progressing so far and if anything needs to be adjusted. We would also move the task that were finished from the previous day from “In progress” to “Done”. If everything was to our satisfaction we would end the daily scrum by assigning each other by preference into groups for pair programming. When the groups were formed, we would start working on developing the product and repeat the same process again in the next day.

Sprint Retrospective

Every week when preparing for the sprint review we would do the sprint retrospective. While preparing the presentation for the upcoming day we would ask ourselves “What went wrong this sprint?” and “How could we improve it for the upcoming sprint”. One of the early issues that we encountered was that we were not fully utilizing the sprint task board, that meant there was confusion in the team on who was working on what. It resulted in two different groups working on the same task, thus creating a conflict in the version control and losing valuable time. We solved this issue in the sprint retrospective by agreeing that when someone works on a specific task, it is properly labeled “in progress” in the sprint task board.

During our sprints we heavily relied on the three artifacts from the Scrum methodology.

Product backlog

Arguably the most important scrum artifact was created at the start of the project. It is typically maintained by the product owner but in our case we were the product owner. This meant that we were responsible for the maintenance and prioritization of the user stories. After creating all the user stories we could think of, we would add them to the Product backlog. The next step was to prioritize them. We were assigning them a number between 0 and 99 based on the importance of the user story. We did not add much user stories throughout the project because we included everything important at the start.

Sprint Backlog

At every sprint planning we would create the sprint backlog based on the contents of the product backlog. It would consist of a user story with the highest prioritization plus the ones that we felt would make a good combination and sense to include. Apart from sprint planning we would always return to the sprint backlog during daily scrums to see if we need to add a new user story or move it to the next sprint.

Burnup Chart

During the sprint we were using the Burnup chart which was automatically generated through icescrum. We were tracking tasks rather than points which in our case told us more about what other developers are coding and how well we are doing as a team. We had some days where we would plateau and knowing this information we should have acted accordingly to the situation and adapt the sprint backlog to any delays from our end. This means that we did not utilize the burnup chart to it’s full potential. We rather used it during the sprint review then using it throughout the whole sprint as a guideline. We also used the Burndown chart in the last sprint.

Sprint task table

This extra artifact was very important for us. We were using it on a daily basis. If a group member would take a task, he would mark it as “in progress”, this way we could easily track who is working on what tasks, to avoid two separate groups working on the same one. When tasks were finished we would marked them as “Done” during the daily scrums. When a task was not finished until the end of the specific sprint we would move it to the sprint task table of the next sprint.