Version control

Version control helps the development team to manage their files. It is a powerful resource, because it is able to track every change in the code and stores a lot of useful metadata about each file - its history, individual changes and by whom were they commited, etc.

Compared to the previous semesters where SVN was our version control choice, in this project we were using git. Compared to SVN, git is not a revision control, it is source code management. Also, contrary to the centralized SVN, git is distributed, so there is not the risk of a single point failure. Git is widely used especially in the open-source community and it has helped our development process tremendously.

One of the main concepts of git is branching. The name "branch" presumably comes from the file tree metaphor and helps the development team to work concurrently and independently on each other. Once each member is done, he "merges" his branch back to core branches which make the actual source code.

There is the "main" branch, which always has working, production-ready code. In the first few sprints, we only relied on this branch and some feature or bug fix-related branches which were used briefly and later merged to the main. Later on, we discussed the option of having another core branch - development, which would carry working, untested code that if stable one day, would be merged to main. We also decided to improve the naming standards of each branch, so just from the name, one could tell what purpose it served. Was it to fix a bug, or to roll out a new feature? It was the norm that feature branch can only be merged to the development one, and we had hoped that bug fix branches would serve the purpose of maintaining the code in main, but it must be noted that we did not deal with this perfectly. Branches including bug-fixing were usually merged back to development to make our workflow a bit smoother. This, however, would not be possible if our code in main was already in hands of the customers who would not have the time for us to roll out a new feature including the bug fix.

In conclusion, using git was a great experience. One can say that its learning curve might be too steep for some, but once learned, git is a essential part of the developer toolbox. What gave us the most headaches was traversing the different branches, merging them and moving back and forth. Towards the end of the project, we all felt much more comfortable and were able to achieve most of the git related tasks without the help of any internet search engine.

(SVN vs Git - Javatpoint, n.d.)

(Driessen, 2010)

(What is version control | Atlassian Git Tutorial, n.d.)