

1. What went well

a. Team and Collaboration

Refer to aspects and specifics of teamwork and collaboration within your team and/or with the customer, which went well. You may expand on examples about successful role assignment, good team spirit (e.g., a member lost their spark at some point and the team spirit brought them back), good communication with the customer, etc.

Our team did a good job of adapting to members having other responsibilities outside of the project and picking up their part. That would not happen if there wasn't good communication between the members. The role assignment was natural and people got to do the things they felt the most efficient and useful at. There wasn't any arguing about who should do more or less, or debate over who should work on which task. The meetings with the customers were efficient and useful. The customers also gave us sufficient feedback and answered all of our questions that emerged.

b. Project development process

Expand on what went well in terms of the different stages of the development process of your project (e.g., did your elicitation processes give you efficiently and correctly the requirements?) See Note 1 below referring to the development process.

What went great was that everybody got to work on everything and gained a lot of useful experience developing an actual real-life web application.

2. What should have been done differently

a. Team and Collaboration

Refer to aspects and specifics of teamwork and collaboration within your team and/or with the customer, which should have been done differently. Possible examples may include (these examples are not complete):

- Better and more frequent communication with the customer; e.g., let them know more often about the progress of the work, agree for more dates of version delivery, etc.
- Distribution of workload to each member
- There was some lack of motivation at some points, so more meetings were necessary to raise the spirit

We all thought that a little more communication and team meetings would have gone a long way. It was often the case that members were doing their own thing and not collaborating or explaining what they had made to fellow team members. Also, spreading the workload more evenly throughout the iteration would have made things a lot less stressful as well. Oftentimes we didn't do much the first 2 weeks and pushed everything to the last week of the iteration.

b. Project development process

Refer to what should have been done differently in terms of the different stages of the development process of your project. For example:

- Break use cases into parts, as some of them had a workload for more than one iteration.
- A particular part was delayed to be delivered, because it was assigned all of it to one person

See Note 1 below referring to the development process.

Because none of us had experience with developing a project of this size and high stakes, we made some mistakes in choosing the right environment and planning the codework. None of us were used to having to plan our coding or writing out detailed requirements and use cases and then writing the code based on that. We didn't expect the grading to be that strict either. The CI environment was a disaster in itself. From choosing the right CI environment to working with it, we gained a lot of useful lessons through failure. Linking issues to commits would have also made the repository more complete.

3. What we learnt

Look back at the last three months and try to understand what you have learnt along the way. What you learnt may relate to:

- Skills acquired related to the development process (e.g., for use case modelling, continuous integration, version delivery)
- Collaboration and communication within the team and with the customer
- Time management (incl. estimation) skills
- Other skills
- 4. Customer's opinion
- Present your customer's opinion about what their plans are regarding the product after the course finishes (you may ask your customer to send you a relevant e-mail)
- You may also expand on your future intentions, in this section. Examples: collaborate with the customer, fix some bugs, implement some extra/desired functionality.

The biggest thing we learned is planning coding among a team of people and not just individually. Planning how much time it will take to develop some kind of functionality was also difficult at first but we got better at it throughout the iterations. Using .tif maps and different geojson files that were specific to our project was also a big lesson for us. Communicating efficiently with the team and the client also got better iteration to iteration in our opinion. It also had to do with getting to know people better.

The client intends to further develop this into an application that themselves and their fellow hydrological engineers and city planners could use for their everyday work. They also asked if any of us would be interested in making the finishing of the application part of their bachelor's thesis. To make the website user-ready, it would need some further work on the inputs the program receives. Right now, the height map that the application uses isn't reliable 100% of the time due to specific issues related to how the height map is made in real life. However, we tried to make sure that the program would be ready to use once that is done.

Note 1:

Development process stages:

- Requirements Engineering: Elicitation (interviews, personas, questionnaires, etc.), Analysis & Specification (e.g. Features, User stories, Use cases)
- Design (mockups, databases)
- Coding
- Testing

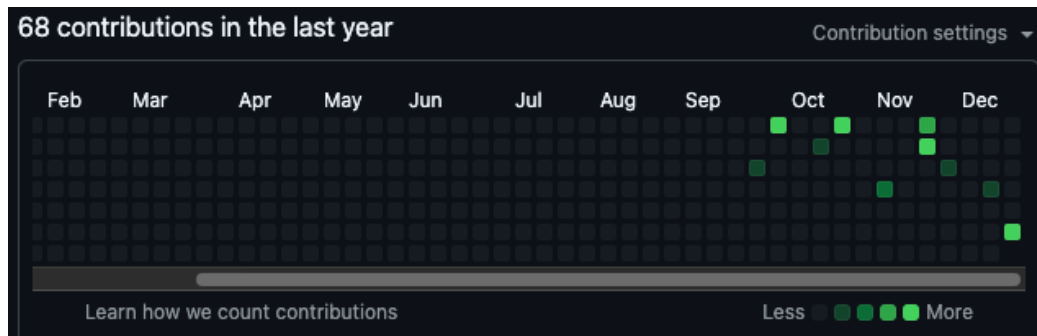
On top of the development process, you went agile, with characteristics such as Iterations and Frequent Delivery, One stage feeding another (e.g., your use cases fed your mockups and vice-versa), etc.

Note 2:

All the above content of the project summary should be a result of teamwork. In addition to that, each member should provide a separate summary of his/her contribution to the project, related to the different stages (requirements, coding, testing, etc.) of the project - this summary could be supplemented with screenshots (e.g. showing info about commits). The member summaries should be included in one appendix in the project summary document.

Appendix 1 - Member contributions.

Ott Eric Ottender - I was heavily involved with the requirements engineering, documentation, planning the scope and UI, and keeping the repo up to date. I was also the main link between our team and the client by setting up customer meetings and clearing up any questions that arose throughout the project. I also participated in the programming of the application by developing the Search By Coordinates UC, Export Data UC, user interface and writing the code for the necessary hydrological calculations that the program does. Additionally, I delivered the presentation for both intermediate and final demos.



Robert Ivask – In the first iteration, I developed our initial two use cases and explored various methods for displaying maps on web pages. In the second iteration, I configured Flask to enable data transfer between the client and server backend and contributed to creating the UI mockup. In the third iteration, I implemented end-to-end (E2E) tests to automate web page testing. In the fourth iteration, I updated the UI to align with the mockups and resolved a few bugs.

Rasmus Meos - I first developed the backend core functionality for rendering the watersheds and river networks using Pysheds (`delineate.py` for interacting with the app and `delineate.ipynb`, a demo file, for visualization).

I then configured the frontend to send coordinates via POST request to the Flask Server. The server would then call the subprocess `delineate.py` which parsed the coordinates and used them to generate GeoJSON files. A script called `convert_geojson.py` would convert the generated files into appropriate coordinate system, making them usable to the frontend. Those converted files were then used to render the layers in the frontend with Leaflet. I also created the sidebar to display various information about the generated watershed like coordinates, watershed area, etc. Moreover, I added various frontend checks to ensure that user-entered coordinates were valid for backend processing.

In the later iteration, I created a Dockerfile to run the app in a container. Once Lauri had created the `docker-compose.yml`, combining the main service and the database service for fetching land usage areas (shapefiles), I improved the functionality by implementing a feature for calculating the land usage proportions and sending them to the frontend for display.

Before the last iteration, I refactored the frontend javascript (which had grown quite large and messy) by separating them into distinct files, each serving a certain purpose. I also made the

sidebar interactive by making it possible for the user to toggle between layers and information displayed. In addition, I added an auto-zoom functionality which focused on the map once the layers had been generated.

The latest additions include integrating drainage calculations with the `delineate.py` script, enabling these metrics for export as a PDF, and creating an additional check in the frontend for verifying user-entered coordinates by water resource proximity. However, at the time of writing, the latter feature isn't compatible with the backend processing yet and is therefore excluded from the final release.

Lauri Kuresoo - I did a lot of research on the possibilities of creating such an application and what kind of tools are out there. I also created the database for storing geographical data in the backend using a Docker container. When creating the container, it also inserts data from the web to the database. I was also involved in creating the UI for land usage data. The land usage layers had to be shown visually on the map and there had to be also a table of land usage percentages out of the catchment. As for the rest of the team, I can say that Ott took most of the responsibility for the progress of the project and kept us on track. I also noticed that Rasmus put a lot of hours into the project and the result shows. Without him we would have a lot less work at the functionalities of the web page done.