ENGR 302 Final Project Report

Project 10: Connected Worlds
Client: Matt Plummer
Date: 19 October 2018

Contents

Project Objective

The objective of the Connected Worlds project was to modularise and improve the existing prototype given to us at the start of the year. We were to also tasked with expanding its usability in an effort to make it more open source friendly.

Summary of Project Results

As a result from our work on the project, it is now in a state where future development can take place more easily. This is due to our efforts to move the project into a React framework and setting up continuous integration to ensure there is always a working version deployed for the client. We also managed to improve the existing product by adding various features such as generic data, year/version selection, and fuzzy search. By introducing generic data we have expanded the domain in which the tool can be used by as now other data sets will be able to comply with our application.

Original and Delivered Scope

Scope of the Delivered Project

In the first client meeting, the scope of the project was defined as there was confusion with the project brief on the ECS web page. The original goals for trimester one were as follows:

- Modular approach to the existing code.
- Increase robustness through adding tests.
- Make project deployable with CI Pipeline.

In the first client meeting of trimester two, the scope was expanded to the features we proposed in the MVP presentation. In addition, our client proposed additions for the scope. The extended goals were as follows:

- Modify data compatibility from only accepting domain specific data to generic data.
- Modify data visualisation layouts to work for any type of data rather than only domain specific data.
- Add ability for end-user to switch between colour schemes
- Made the project auto deploy via either GitLab CI or CircleCI depending on the git platform used.
- Add ability to show data based on year range.
- Add tags to nodes which can be searched for.
- Add direct URL for each node.

Specifications of the Delivered Project

The delivered project has all features from the trimester one scope. However, we missed a couple of features from the trimester two scope. These missed features are as follows:

- Add tags to nodes which can be searched for.
- Add direct URL for each node.

The reason why these features were left pending was because of time constraints. We were given a significantly larger scope for trimester two and the team made a decision to prioritise other features which were more important and realistic to complete by delivery. On the contrary, we did leave behind information regarding how to tackle the remaining two additions. Therefore, in order for any future teams to pick up these additions, they will need to read through the technical documentation.

Original and Actual Schedule

Our original schedule only specified hard delivery dates based on the ENGR301-302 deadlines. This was because we were performing a large refactor in the first trimester and had a poor idea of what we could get completed over the year. The actual schedule was vastly improved in the second trimester when we were adding features to the project. This still fit within the original schedule but had more depth to it as we got on top of our management overall.

Delivered Expenditure

No expenditure.

Project Self-Assessment

Considering the state that the team received the project, we did a superb job in respect to the technical aspects. Originally the code base was created by a Designer with minimal experience working with JavaScript and code in general. This meant a bug prone and difficult code structure, no testing and poor code practises. The team exercised their Software Engineering knowledge to ensure the product was delivered to a standard we were proud of. This meant reworking and rewriting the code, adding in a testing and implementing a CI service. This was evident through linting our code, putting the work into an industry framework, utilising CI early in the project and cleaner methods for future developers to understand. Furthermore, our client and stakeholders were pleased with both the documentation and quality of our code. This brings us additional confidence in the final product with respects to the quality of the technical aspects. It is great to deliver these to a high standard considering our client previously had received work of a poor quality, no testing framework and no documentation.

Lessons Learned Summary

Learning about management has had a high significance and impact on how we as a team have worked on project and how we have worked together. At the start of the project we did not put enough focus on following a proper sprint cycle with a strictly adhered-to retrospective and planning for what to do next. This led to confusion on what everyone was doing and meant some features slipped away due to poor communication when they should not have. Upon reflection at the end of trimester one, we introduced a more structured process from Agile, two week sprints with meetings at the start of each to communicate what needed to be done and who was going to do it. Because of this the process of implementing features improved in respect to quality and timeliness. Thus, we have learned to appreciate the importance of a strong project management culture and organisational structure. As an added bonus, we actually enjoyed all the administration work that came with this project which increased the engagement and group commitment to the project.

Furthermore we also learned the importance of implementing retrospective meetings and documented procedures for operating within the group. Not having had these originally meant that we had significant issues with the distribution of information through our group. Because of this before we implemented them development was often slow and difficult, forcing team members to break the flow of work to ask each other questions and spend a lot of time finding answers. It also meant that originally it was difficult to work outside of the area you had started in and getting help meant bring someone up to date on that entire section. Having implemented this solved all of these issues and allowed group members to voice concerns and congratulate each other on good work. When this was combined with a strict procedures document that we updated using these meetings it heavily increased work satisfaction, quality and most of all speed.

On top of this, we learned the importance of managing documentation. The lack of documentation in the prototype significantly impacted the early stages of the project. The messy and unclear code of the existing prototype served as an obstacle to every member, as we had to tackle this alongside having the bulk of our team learning React.js and JavaScript for the first time. Having to do this without any supporting documentation added significant difficulty to the early stages of the project. Moreover, the addition of our own documentation helped us reference how we had done the technical aspects. This act of management helped lighten the load for the closing stage of the project while assisting us in implementation.

Looking back at the year, there were issues that could have been mitigated earlier had we had kept on top of risk management. We had taken the first steps in risk management by identifying and outlining mitigation strategies for various risks for the project, yet we did not follow through by reviewing our current situation regularly. A common downfall was our estimation on development times which we acknowledged as a risk. While we took small steps toward fixing this issue in our retrospectives, it would have been faster to fix this should we have looked into ways to improve from other sources online rather than reinventing the wheel. A result of this was a fall in our development efficiency. We have learned that it is important to keep on top of risks as these have a waterfall effect on the rest of the project should they not me maintained.

Procurement Summary

No procurements.

Transition Plan

Procedures

We will transfer our version of the connected worlds project over to a git repository provided by the client and deploy it for them. We will be doing the steps outlined in the GitHub section of the Setup guide. This will contain a directory containing all the required documentation for the continued development of the project, the administration of the project and the use of it by any member of the public. There will also be a meeting between a group member and a stakeholder to answer any questions not answered in the documentation and to facilitate a smooth handover of the maintenance and administration of the site and data.

Resources

No required resources.

Assumptions

We have assumed that the client has access to one of GitHub, BitBucket or GitLab to host the codebase and data on and as a base to run the CI from. We also assume that the client has access to either GitLab CI or CircleCI (depending on git platform) to run the tests, parser and deployment scripts from. Finally we assume that the client has access to an administrator who has enough technological knowledge to use git to access and change data, do minor troubleshooting of the website and follow the setup guide if needed. Finally, we have assumed the client will be able to host the application on a static website such as surge.sh.

Organisation

Organisational requirements of both parties for project transition to be successful consists of technical and non-technical support.

Technical Support

We must effectively pass down crucial technical knowledge of the project to our client and his stakeholders. We will provide this support through a closing meeting to walk through the project from a developer's point of view. This closing meeting will also support the hand-over process by walking through the steps to set-up our work into the client's workspace. Moreover, we will create a demonstration video as requested to act as a quick reference of how to work with the project at a technical level.

Non-Technical Support

We will provide all the documentation that is significant to our client. This includes the set-up guides, installation guides, user documentation and technical documentation.

Work Required

- Cull stale branches
- Merge remaining branches
- Include GitLab issues to explain how to finish the missing features.
- Polish all documentation
  - Final project report
  - Installation guide
  - User manual
  - Demonstration video
  - Architecture Documentation
  - Technical documentation

Schedule

On the 18th of October 2018 a team member will meet with a stakeholder, Andre Geldenhuis, who will be assisting Matt Plummer with management of the site once the transition is complete and explain how the site works to him.

On the 19th of October 2018 the team will send the completed documentation to the client. We will also set up the site on a Git repository for Matt Plummer on that day. The documentation will be zipped and emailed to them by 5pm. The project is complete on this day.