
1 WHAT I LEARNED: EVAN TSCHUY

At times fun, at times incredibly frustrating, the last year of capstone has been a great opportunity for me to get a look at the perils of project management, and dig into creating a technical product from nothing.

1.1 What technical information did I learn?

On a technical level, the project can be broken down into several sections:

- a NodeJS backend:
 - direct interactions with ElasticSearch
 - a custom-written library for interacting with `git` repositories
 - 3rd-party authentication integration with Google
- a frontend:
 - a single-page application using Aurelia
 - Materialize for user interface items

Every single one of these sections was its own learning experience for me. In the past, I have written web applications for work. These generally tended to be fairly self-contained, with simple user interfaces that would fit in well with the year 2009.

Instead, this project was hyper-modern. We purposefully chose a single-page application backed by a modern backend, a complicated datastore, and a powerful, real-world search engine. All of these things were new to me, and being exposed to them has prepared me in the event I decide to continue creating web applications.

1.2 What non-technical information did I learn?

Some people will need pushing to get things done. There's no way around that, and so it is simply necessary to know when to push and how. It is necessary all throughout the year to make tradeoffs in who does what, when, and why.

1.3 What have I learned about project work?

It is easier if you just accept that some people put in less effort. After weighing how much I wanted to finish the project and how much recognition I would get from my group, my peers, the school, and others, I simply went ahead and did parts that we had earlier divvied up otherwise. The other option would simply have been to let the project go uncompleted, which I was not alright with.

1.4 What have I learned about project management?

I think that what I learned about project management can be summed up in one small line:

Divide up work early, and check progress often. Readjust as needed.

1.5 What have I learned about working in teams?

Working in teams is a give and take situation. For parts of a project where you are the most technically familiar, it is often a good idea to take on that section while someone else works on things they are more familiar with. On the other hand, if you have a little more free time, it can be good to instead work together on both sections, so that you can learn what they know and they can learn what you know.

1.6 If you could do it all over, what would you do differently?

I have told every single one of my friends that are going into capstone next year the exact same thing: **make a team and contact a good project, and do it — don't just do the rankings on the website.** It is worth it to not get stuck with strangers on a project that you didn't choose, and you might get stuck with a project you didn't choose even if you submit your rankings. That has lead this past year to be much worse than it could've been, had I just had the energy to put in to corral some of my friends into doing a project we liked. It was an alright year but it could've been much better.