Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

CSU33012 Software Engineering

Individual Reflective Essay

Liam Junkermann, 19300141

November 30, 2022

This module focused on teaching us how to work in teams. The assignments were all fairly straightforward tasks, aimed at highlighting the collaborative steps and strategies to effectively work in a team. This reflection will focus on the learnings and effectiveness of our team in the main collaborative project.

I have a strong background in building, packaging, and deploying web apps at various scales. Therefore I fell into a pseudo-advisory role within the project. As a group we met at least twice a week to work through the project together, as well as being in constant communication through a discord server. Through both these approaches we were able to brainstorm, build, and iterate the idea. Through each of these phases I offered my experience to help inform and make decisions on the project scope, approach, architecure and solution. In group coding sessions I was able to do research on API usage and how best to implement or offer the data, as well as providing support with coding the solution based on my experience.

Most of my experience, though, has been through work done as a single programmer. Learning to work with multiple people on a project from scratch was a challenge I was not expecting and have since learned from. Our repositories branch structure has room for improvement as a result. The steps of experimenting with different apis, to determine the right solution for us, and combining the learnings from each api subteam into the final solution were slightly disjointed. So there are some branches which have competing and unrelated branch histories which have been manually migrated to the final solution branch.

Our teams journey to the final solution was far from straightforward. THe first task was determining what metrics we were going to try to track. This decision took two meetings across multiple days. Once we settled on comparing weather with commit counts, we began exploring options for where to source data and what our architeture would look like. At first we looked at using Java all members of the team were familiar with the

language from college. Webapps built in Java tend to use the Spring boot framework. Setting up springboot, and more importantly, reliably running springboot across everyones different machines and architecures proved difficult so we began exploring other options. Finally we settled on using NodeJS and react, specifically NextJs. NextJs is a web development framework built on React which offers server side rendering for static pages, as well as api functionality allowing us to build functions and api endpoints to consolidate data retrieved from multiple APIs into one request which our frontend can leverage. We decided to use the typescript flavour of NextJs to provide additional reliability and predictability to the solution. Typescript provides static type-checking, not only making intellisense more effective in the IDE but also alerting the developer to any potential mistakes in their code before it is committed, or worse, shipped. Typescript also makes it easier to follow what a React component might be doing and where the data is coming from and what the structure looks like. In a project like this where so many different data sets and datatypes are being collated a tool like typescript is invaluable for the team in speeding along the development process.