Name: Jadon Procter

Term: Spring 2022

Previous Team Projects

I have worked on other team projects, but mainly with just one other member and not with a continuous integration - except one project. For my databases class I had to work with another developer on making a react site linked with a SQL database. During that time I learned loads about Git and how to avoid merge conflicts and such, but didn't know anything about pull requests and things. My role was that I had half of the database to configure, half the api that talks to the database and half the react website. The major difficulties on that project was the merge conflicts from not understanding git and github remote repositories. I learned to pull my weight but also work as a team and reach out for help when I needed it, and give help when it was asked for. Other than this project other team projects have been where each person is able to do a part of the project that is entirely separate. For example in software development one we had to create microservices in a team, but we each worked on our own microservice which meant we didn't need collaboration till the very end.

Working with Continuous Integration

I was nervous about continuous integration (CI) as I had merged conflict issues with a previous project that set us back lots. In working with CI I learned how to make a pull request which was super helpful in the process. I approached writing pull requests as making sure I was detailed in my commits and consistent in the TDD process. I tried to make sure that the pull request was as up to date as possible before the code review happened. Writing code reviews I found to be a helpful way to understand how someone else codes and just because it is a different approach doesn't mean it's wrong. Learning how others approach problems helps add "tools to your tool belt" in terms of approaches. I ended up finishing my portion of the project early which freed me up to help my team. In this I tried to commit early and often. After working with this team I have grown as a developer by understanding how to code with a team and use pull requests / git to avoid merge conflicts. Working on my own branch and making sure the file layout works in a separate way to avoid merge conflicts. As a team member I grew in terms of trusting my team. I like to finish things early and some people work to the deadline. As a mentor I believe is where I grew the most. I was able to help my team when they were having trouble and almost pair-program / guide them in the right direction when they were stuck.

Lessons for the Future

CI has shown to guide better software development as it makes the process extremely collaborative and forces individuals to build team skills. It also protects the code from large commit sizes that change all the code very fast, and forces developers to commit early and often. One part of CI that helps steward collaboration is code reviews. Development is a constant learning process and code reviews are one of, if not the best way to learn from the code you and your team writes. I found that along with CI and code reviews, test driven development (TDD) is a good thing to incorporate with CI. TDD allows developers to commit early and often, and gives them a goal to work towards. Having a goal to work towards allows the small commits to actually be very efficient in the code written. One tends to put quality over quantity in TDD. With TDD comes the test suite. Having a good test suite with a shared repository adds the ability to collaborate on tests and thus catch more bugs before production. Overall I feel that this assignment has made me a better developer in actually understanding and doing the software development process from beginning to end.