Software Engineering Group Projects: Reflection - Individual

- To what degree were you able to follow an iterative approach and deliver some working software with valuable features for your client at each end-of-iteration checkpoint?

  I think the nature of a web-app makes it easy to iteratively develop features since we just wire them to new endpoint; for the first three checkpoints I think we added a ton of features in between each iteration which led to our client increasing his demands. Towards the final checkpoint, it was more important for us to refine what we already had and expand upon existing features.

- Looking back at your original project plan, what changed, and why?

  After checkpoint one, we ended up shifting around our objectives for different checkpoints, as well as removing some and adding other to meet our client's specification. An example is search, which we intended to have by checkpoint three, with advanced search features implemented by the final checkpoint. Search became something we set aside until the end because we had to move our resources to
  I think one key change was our MVP, which kept on evolving throughout the course of the project. We were originally meant to have an MVP ready by checkpoint two, but it ended up taking almost twice as long because of these changes and some technical issues.

- What technical software engineering practices did you apply during this project? Which were the most valuable? How did they help?

  We initially intended to develop this app using a scrum ideology, and vertical slicing to keep our group flexible. I personally quite liked the scrum boards because they were straightforward and concise.
  Vertical slicing is a good idea, but I think in this case it actually hindered our productivity since most of our group came into this with almost no experience working with reactJS and the Flask framework. What happened was that from the very start we each focused on one side of the project, and it made it really difficult to understand what was happening on the other end.
  I started off with the backend and did a lot of the work on that side, so when it came time for me to work on front-end development, I had to spend far too much time trying to learn how it all worked. I would often have to ask certain members in the group how to do something trivial, and also found that it went both ways. Certain group members who

specialised in the react side found it very difficult working on the backend. They would get stuck on a certain bug for hours at a time which I would fix within 5 minutes of looking at the code, which is clearly a very inefficient way to divide up our time.
These issues could have been avoided with better communication, more experience in the languages, and more pair programming sessions, but I think it would be very difficult for us to do much better considering how busy everyone's schedule gets during term time.
Another practise which I personally tried to use was TDD, particularly on the backend where I did the majority of my work. TDD really helped to home in on bugs, and to refine our API, however it took a long time for me to learn how to test in python effectively. I think it's a shame that the rest of the team didn't even try to write tests.

- What were the main things you did in order to improve your team's productivity over the course of the project? What led you to take this course of action? Did it work?

  I had to help a few members of the team do a lot of their backend tasks since they were unfamiliar with it. I also felt like I had to take on a lot of heavy tasks, such as implementing live website updates, because the work wasn't being done. There were times, especially toward the end, when team members wouldn't really spend their time working on this project.
  I also spent a lot of time refactoring code for clarity, which I feel did have a slight impact on our own internal workflow, but I think it will be particularly useful for other people looking in.

- What advice would you give to another team undertaking a project like yours in future?

  I think it's really important for groups to maintain good communication at all times, and to have regular group coding sessions. It's also really important to adhere to the original plan because it can be very easy to stray from it, and to divide up the work evenly.
  In regard to a web-app like ours, I think the very first thing that groups should do is to create an abstraction for the components and features of their website and decide what kind of API they use. Naturally things change during the lifetime of these projects, but I still think that having a general idea of what the end goal is, and how things interact helps improve output.

Moreover, I think TDD is a very good practice which greatly helps with debugging, and I would encourage everyone to use it.

- What could be improved about the course to better support teams working on their projects?

  It should be easier for groups to arrange SE consultation meetings, and they could be a lot more useful. For a lot of these projects, students are going in inexperienced, like for our group where maybe only two people had worked with the technology we were using. And even though a large part of the reason why we chose to do this project the way we did was to learn new skills (Flask and React), there were a lot of pitfalls along the way. I'm sure that like our group, many others had this same kind of problem, so it would be really helpful to have these in person meetings more often to prevent hours of reading stack overflow.
  The number of hours we spend working on these projects often far exceeds what the college actually expects of us because of this, and it ends up discouraging many group members from contributing, which leads to a kind of snowball effect.
  I also feel like I don't really understand what the role of our supervisor is. I initially thought he was a kind of personal SE consultant as well as the person assessing our project, but I don't think anyone in our group really went to him for help. In the end we only saw him at checkpoints where he would give us a grade, so it felt like he was more of an overhyped marker than anything else.
  For the future the college should clearly explain to groups what their supervisor can actually do for them (not just as a one off at the start of term). If it is just a marker, then maybe extend that role to help the group, and give detailed advice. The help our supervisor offered us was basically suggestions for features without explained how to even start going about it – it was kind of like "this thing should be done, go figure it out" rather than "you can implement this, you should maybe look into x library…".

- What would you like to do better in future projects (either in College, or in industry) - or what could you take from this project to usefully apply in future?

  I am actually really glad I did this project because I feel like I have learned a lot of valuable SE skills. I've become a much better Python programmer and I feel very comfortable with the Flask framework. I have a much better understanding of websites – in terms of security, sessions,

and memory management which I think will be extremely helpful for personal projects in the future. I also learnt some basic JavaScript, specifically reactJS which I would like to use for more projects.
Overall I feel much more confident as a web developer than at the start of this project two months ago and I think I could go on to build my own websites pretty easily now.
In terms of a reflection, and what I would do better, I think I would encourage more TDD, and to be smarter with organisation. An example of ‘smarter’ would be to assign tasks to team members who specialise in that area (or have vertical slicing where it is actually appropriate). I think I could also greatly improve my communication skills, so that’s something I hope to work on.