

During semester one in Computing Science level three, we were required to work in teams to complete a group project for the “Team Project 3” course. More specifically, we were required to create a Drug Calculations learning app for the School of Veterinary Science of the University.

Ms Fiona Dowell, a lecturer in the School of Veterinary Science, found that her students struggle with learning how to do drug calculations more than they do with any other part of their course. She came up with the idea of having an application available to Veterinary School students to help enhance their drug calculation skills. Ms Dowell believes that students will find it more entertaining to have a game-like learning application, which they could use in their free time to entertain themselves, but also learn and become better with their drug calculations.

We decided to try and get hold of our client for an interview, in order to gain more information about the requirements and specifications for this application. Fiona was more than happy to meet and discuss with us her thoughts and ideas. We then had to work as a team to analyse any data collected from the interview, create a project management plan, split roles and work together to achieve our goals and deliver the products to our client.

We carefully created an interview plan and collected our data from the interview. We then had to carefully analyse our data and try to understand exactly what our customer had in mind before starting to actually develop their software. What Fiona had in mind was an application which would mainly have 2 types of users. Administrations of the application, which would be able to add or modify material from the application, and students who would only be able to access the application to view slides and answer questions / complete assessments. The general idea was that a student can start by viewing a number of slides for a certain topic and after he/she feels comfortable enough with it, go on to try out questions based on that specific topic.

The team project course gave us more freedom for the software development model we could use so we decided to go with the Agile Development Methodology, which was defined in Section 4.6 of our Professional Software Development course notes; This allowed us to have continuous communication with our client and have the client involved in the development process at all times. What this offers us is not only greater flexibility through the development process, but also making sure that the client will be happy with the end product as we are getting their feedback for them continuously.

First thing we did after collecting our data was to create some paper prototypes to make sure we are designing exactly what our client had in mind. We then consulted our client to see what they thought of our prototypes. Afterwards, we created some user stories, which represented actual functions that we wanted our application to be able to perform. An example user story would be “as a lecturer I want to be able to add new learning material to the application”, which can be then translated to a task called “add functionality to allow adding new learning material to the application”. We decided to use GitHub for our revision control and even though we used to keep our documentation on Mahara we have moved that to GitHub too. A member of our team had previous experience with GitHub and after a discussion with our supervisor, we were convinced that learning to use GitHub would be a worthy investment not only for our current project but also for our future careers. GitHub offers a number of advantages over other revision control systems. For instance, with Git, each developer gets a local copy of the repository and can commit to it even when he/she is offline. When connectivity to the main repository is restored the developer can then commit all their local commits against the main repository. Moreover, GitHub enables us to keep our documentation, code, wiki and issues in one central space where all team members can access and modify data from any computer with no software dependencies.

Some difficulties came up during the course of first semester, such as the fact that some of our team members were difficult to get in contact and everyone could easily get confused about who's parts are completed and who's parts of the application are still in progress. To resolve this I suggested we start using the Issues functionality on GitHub. I have explained to everyone why I believe this would help us increase productivity within our team and after everyone agreed we started using it. This allowed us to create and define tasks that needed to be completed and assign them to a team member. We then have a tester who is responsible for testing everything that is marked as completed before an issue can be closed. This can help keep everyone aware of what they need to work on but also make sure that whatever is completed is also validated by our tester.

There were times when a member of our team would take down the whole project by overriding everything on our repository with an incorrect version from their local copy which was happening due to inadequate training of that person on github. To resolve this issue we arranged a team learning meeting which included a small training session such as a summary and good practice rules for version control on GitHub.

After all our data was analyzed and our requirements documented we had to decide on which framework we would work with to create our application. Not all of us had experience with web development so no strong preferences towards a specific web framework existed within our team. After doing some research on web frameworks we came down to the best three. These were Web2Py, Ruby on Rails and Django.

Our supervisor suggested that we get three members of our team trying to create a simple application each, with one of these three frameworks.

I was assigned the Django application, so I decided to go through the official Django tutorials and see what I can get done. I realised that the tutorials Django was offering were taking you through the best tools that Django has to offer, so that whenever you need to create your own application you will have the experience required to get the basic functionality working and then you can always use the online documentation and support forum for anything advanced.

I demonstrated my simple application to the team and explained how my learning experience using the online support resources has been with Django. After comparing my application and learning experience of my Django application with the applications of my other team members, we decided that we will use Django for our application for a number of reasons. One of them was the fact that it uses Python code, whereas Ruby on Rails uses Ruby code. We have been using Python for the whole year in Level 1, so that was a clear advantage for us. Secondly, we found that there is better documentation and support for Django than there is for Web2Py and Ruby on Rails and that would potentially help us in the future with any issues we might be having with our project. Another important factor was that Django offers a really nice way of creating an administration page by only providing the models (objects) and some additional information for them, which means that we would save ourselves a huge amount of time that would have been otherwise spent writing the admin site from scratch if we were working with Ruby on Rails or Web2Py.

We then went on to create a Django prototype for our Client, which will soon be ready to be sent to our client. Our client will have the opportunity to work with the prototype and get a general feel of how the application will function before getting back to us with either positive comments or with suggestions we can use to improve the product we will be building later in semester 2.