

Management report

Group No. 2

April 6, 2016

1 The project planning process

All of the project planning was done via the weekly meetings, which were found to be sufficient. Each week we would review our goals and set intermediate objectives. The team members would then choose out of the pool of available tasks, concluding the meeting. It's an extremely simple but a practical scheme, echoing agile development.

The weekly meetings were essentially perfect for the scope of our project. A week's time was sufficient for implementation and development of most features and foundations, whilst being occasional enough not to bog down the team with excessive micro-management and unnecessary meetings.

Ultimately, the project simply did not require any more than a week's foresight, since each week we did what we could to further development, and an overall plan was not going to change that metric.

2 Project goals and objectives

Since our project was an idea of our own, we were careful not to attempt to tackle anything too large. As a result, our goals were appropriately sized for our team. On a similar note, the idea, a computer science forum. being our own, was very well defined, and so no changes took place. We did precisely what we set out to do.

3 Project scope

Due to our own experience of cs3013 back in second year, we were very well aware of scope creep. To combat it, as mentioned before, we focused on developing a no bells-and-whistles forum, restricting ourselves to implementing core functionality and taking the time to implement it well, as opposed to rushing it in order to proceed to more interesting and exciting features. This served us well, as we accomplished our goals and built the fundamentals within the time frame we had hoped for, and furthermore, built those fundamentals well.

4 Project approach

As mentioned before, no tools were as useful as simple communication. Our project approach was simple, we were in it together, and did what we could. Since everyone was doing their best, there was no need for greater management, because there is only so much a person can do in a given time period. No amount of planning or discussion will increase that metric.

Each week we would discuss what needed doing next, for example, forum registration needed to be complete. Within the meeting we would then discuss whether the goal, in this example, the forum registration, was too large to accomplish within a single week. If we decided it was manageable, then that was the end of it and we'd have a target for the week, splitting up the workload between the members (i.e. one person would do e-mail verification, another interfacing with sql back-end etc.). If we decided that the goal was too large, it would be split, and the process would be repeated.

5 Project organization

Our simple weekly-meetings only approach worked wonderfully for our small-scale project. Although it is not particularly scalable, it was quite appropriate for us. Each week goals would be set, those goals would always be completed come next week, and the cycle would be repeated. Honestly, it is a testament to the team in question, as opposed to the organization methodology. The combination of a small dedicated team and little-to-no planning overhead or micromanagement resulted in swift, pleasant and stress-free development.

6 Risk analysis

Our core risk, which was identified at the very start of the project, was user malice. Computer science students like to attempt to pick things apart for a variety of reasons, and our forum was sure to be a target. The main form of attack we had to protect against were SQL injections. Thankfully, seeing as our development was smooth and swift, we had leftover time to patch security issues and sanitise user input to prevent such injections. No other risks were identified or came up, seeing as the project was a rather simple one.

7 Project controls

7.1 Scope

As mentioned in the Project Scope section, scope creep was not an issue due to clearly defined goals at the beginning of the development.

7.2 Schedule

Schedule was controlled via the weekly meetings. Seeing as the largest period of time between updates was a single week, no issues could grow to truly stall development. Thankfully, there was little to no actual controlling done, as every member "pulled their weight" without additional incentive to do so.

7.3 Quality

Quality was ensured via peer code-reviews, which is to say people looked at each other's code whilst developing. We have not had any issues with this form of quality control, but then again, the project is small-scale, so it is harder to make big mistakes. However, as mentioned, it was sufficient for us.

8 Communications

The idea was our own, and so in lieu of a client, we spoke with our classmates, who are after all the target audience. It was effortless for us to ask for feedback, as we all had connections within the computer science course. The communication however yielded very little, seeing as the team, too, was the target audience, and between all of the team members we had a very good idea of what a computer science forum should be able to do.