Project plan

Group 3 - Git Gud

May 9, 2018

To execute and complete a project of this size, is something that this group has very little experience with. Since we don't know what to expect in the coming months, it is in our best interest to use an agile programming method. It is difficult for us to plan in a traditional waterfall style when we have never had the experience in working with big teams. This project will have aggressive deadlines and high complexity, which requires us to have continuous planning and continuous evolution of both the project and the software. It is also most likely that we will change our roles and the way we work throughout the project, agile methods are perfect for this.

The agile method we have decided to choose is scrum. Scrum teams does not typically have a team leader who decides how a problem is solved or delegates tasks, rather they have a team who equally decides how to turn product backlog items into ready products. This suits our group well, since we have decided to make all our decisions in plenary. Although we dont have a typical leader, we have chosen a person to have an overall responsibility to structure the group.

Every team who uses scrum needs to have a to-do list, also called a product backlog. For this we use the web-based project management application, Trello. This allows us to keep track of which tasks that needs to be done, who's doing what, and which tasks that has been completed. Tasks from the product backlog will be completed in so called sprints. The tasks on Trello, are so called User-stories. User-stories are small task that needs to be completed to accomplish a finished product. Task will be divided in plenary in our weekly meeting. This is important to get a common knowledge about what needs to be done. A scrum sprint is a time-boxed period where specific work is done. Sprints are usually 2-4 weeks long, which is typically the time interval of an obligatory assignment. This is another reason for why scrum would be a good fit for this project.

We will not be using TDD. The reason for this is that we have to little experience with this as a group. To successfully develop a program using TDD, requires thorough tests before a single line of code is written. Since we dont have much experience of writing good test, TDD is something that would have taken us a long time to do. Therefor we have decided not to use this in this project. We did considered using pair-programming, but as the course has gone by, we have decided not to use it. We have however programmed together in groups, with all members present. Although we will not be using TDD, we will be writing tests along the way.

Another important ceremony in scrum, is the daily stand-up. This is a short meeting (about 15 minutes) where the team members cover their progress since the last meeting and plan their work for the next meeting. In our team, we won't be able to have daily meetings, because we all have different and complex schedules. A solution to this is Discord. Discord is a program where we can have daily chats, much like the daily stand-ups in scrum. In addition, we have agreed to meet every Thursday, 10:15. This way we can have a "show-and-tell". Show each other what has been done, and discuss what needs to be done.

To make the development process easier and more efficient, we decided to make us of planning poker. We would write down the requirements for the sprint and then each member rank the size of the feature. Following the rules of planning poker, if the team agreed that a feature was very large, the feature had to be broken down into smaller tasks. If two or more members have very different measurements of a feature, said members must explain their reasoning, and the team will then adjust the tasks for that feature accordingly.

The first thing we did when we formed our group was to talk about each members experience. We did this so that we could more easily divide the different roles. Only one of us felt comfortable with git, so we gave that person the overall responsibility with organising and structuring of our git repository. We came up with clear guidelines for our commit messages, and the frequency of our commits. One rule is to not commit more than necessary and only to commit when a working change has been made. This way we wont have thousands of commits, and it will be easier to go back to earlier versions if we make a mistake. As far as the other members, we have currently divided the roles of; Scrum Master, chief of programming and chief of graphics, Since we dont have much experience with working in big teams and dont really know what to expect, we wont divide more roles before we start with the actual coding.

Another important discussion point, was to figure out each members schedule so that we could organise a weekly meeting. This turned out to be more complicated than we first thought. Most of us have different classes to attend and a few of the us also have a part time job on the side. To find a suitable meeting was therefore difficult, but at the end we chose to meet every Thursday at 10:15. We felt this was necessary so that we could have sufficient communication and discussion in our group. We will therefore have two weekly meetings, one at Tuesdays with our TA, and the other on Thursdays on our own initiative. Depending on the difficulty as the project advances, we might have more weekly meetings.

With the current combination of planning poker and trelloboard, our team has experienced, and is confident in that all human resources and tools are being exploited to their full potential. Planning poker ensures fast break down and identification of problems and tasks. Trello gives a total overview and easy tracking of what needs to be done and what's completed. Trello also tells the current workload of all team members, thus making it easier to assign urgent tasks and move team members around to tasks that need more work force.

When working with projects of this size there can always be problems and unpredictabilities along the way. It is therefore important to have a plan for how to encounter these problems. Communication is key. As mentioned above, we use discord. This allows us to have good information flow if a problem occurs. Discord also allows us to use screen sharing and video calls, which is important when it comes to solving programming issues. Another measure we use to handle unforeseen situations is to work hard at the start of each new obligatory assignment. This way we have a greater margin if a problem should occur.

References

[1] *https://www.versionone.com/agile-101/what-is-scrum/*, What is Scrum? 2018