Retrospective iOS

During the initial phase of our sprint planning we had a specific mindset regarding how much work can be accomplished during a sprint. As a group we didn't have much experience working agile. Individually we had more or less worked with sprints at different times but never regularly and consistently and not together as a group.

Our primarly struggle at the start of this project was to estimate the time needed to complete each feature as content of a sprint. As reference we used the course laboratory assignments to measure and form a perspective to estimate the amount of time that would be demanded for our features. Arguably our sprints consisted of features that could be comparable to these laboratory tasks, which made them liable reference points for us to form our sprint planning.

Additionally we had a course parallel to this one in which also consisted of a big software project. In the best of worlds, we would preferably all work full time on this project, but facing the situation we had to split our time relatively between both of them. We realized that a situation like this could possibly occur in future work and took this as an opportunity to strain ourselves and create an achievable sprint planning. We did not want to face a situation where we couldn't push a functioning feature at the end of the sprint and in turn extend the workload for the upcoming sprint.

When planning our sprints, we included several different elements considered needed for completing a sprint. Such an element could be time researching material for our different features. Since we at the beginning were not entirely comfortable programming in Xcode we had in mind that the learning process would also consume time, this includes the time we would look for solutions and guides on the internet. So to be on the safe side we accounted these elements as somewhat major parts for each sprint.

It's preferable to finish with the features before the sprints deadline are due, than the other way around. It gives time and opportunity to improve the quality and oversee potential bugs that can be crucial for the user experience. In our sprint planning we calculated to also include some form of testing, in which we would have time to look for and fix potential bugs.

As the project ensued and we completed sprint after sprint we soon realized that we overestimated the amount of work for our sprints as they took less time to complete than anticipated. Certain elements in the sprints and their given time allotment became abundant for the individual sprint. We didn't need a timespan for research when we were implementing features that did not require any research. It also came down to that since we could reuse code and work our way from there, we didn't need as much time to look through guides for every sprint as we thought we would. Another part was the sheer amount of time it would take to write the code. Features that at first thought we would consider taking huge amount of time implementing, in reality just took an afternoon to complete.
This made us finish sprints before originally intended.

We saw however this as an opportunity to use the remaining time for each sprint to tweak and configure our features to extents that was beyond the original planning without adding any additional features.

A feature that we originally intended to use was *push notifications*. With this feature the user gets a "Daily reminder" that can be anything from inspirational quote to a supportive compliment. We wanted the application to be more close to the user, seemingly interactive in a way.

The push notification feature was somewhat harder to implement than we first imagined. In term of code it wasnt too hard, but ultimately the school developer account was not too keen on letting us use this feature. It was not simply a string of code and a few adjustments, but generally a whole other spectrum with essentials we were unaware of. We used our distributed time of the sprint to search for solutions and guides on how to implement this and even though we used this time effectively we still found ourselves short of time.

We re-evaluated the feature within the group and came to the conclusion that it was not worth the time spent. Further research clarified for us that push notifications was not the right way to go. Notifications require relevant and timely oriented data and we did not deem our intent to fulfill any of these criteria.

In hindsight we feel that this feature would be an interesting thing to learn and to use. But we are satisfied with the decision of not letting it consume more time than we originally planned. If this feature was somewhat of a core foundation for our application we'd probably allow it to take the time needed and arrange our sprint planning accordingly. Excluding this feature is a lesson learned from our point of view, as we see all unforeseen event to be realistic and very much possible out in the real world. You will at some point be in a position where you have to scrap an idea that you praise, in purpose of the bigger picture. Learning the value of this and be able to deal with the situation is much warranted when working on greater projects.

We ultimately reckon that we can learn this feature individually in the future, the key lesson was to be able to understand the importance of it and to work around it.

We obviously wanted to distribute our application and make it available to everyone. As an application creator regarding iOS we consider it a necessary trait to be comfortable and experienced with the publishing of apps to App Store. The process can be somewhat rough and disorientating for the unaccustomed and we wanted to familiarize ourselves with it.

During the course we had a lecture where we got thoroughly guided through the process and presented with the different components. We wanted to put this in practice and scheduled it for the last sprint in our sprint planning. We were assured by our course teacher that uploading to App-store should be working with the account we had at our disposal.

When we eventually got to the point where we were about to distribute our application we ran in to a dead end. Our school Apple-account was not authorized for publication on App Store. This was an issue that was out of our hand and certainly not something we could've prevented. On the other hand, looking back at the situation we should have put our self in a situation where we make a genuine check to see if all the elements for uploading to App Store was functioning. However out in the real world you must trust other people to assure that components are in working order, as you cannot take on all work yourself.

If we were to face this issue on a live project we are supposed to publish it would've been more severe. On the other hand we are also aware that certain precautions would minimize the risk of this kind of event. The account in question would be more of a liability to the company since it would've been used on several occasions and initially not create these kind of problems. Also the economical and supportive reassurance of the company you work on would make it an easy fix, i.e. in worst case creating a new company Apple account.

This project has been of great educational value for all individuals in this group. We are very pleased with the tasks we have done and the challenges we have faced. We are aware of the lessons we have learned and their value as we see them as great experience for the future, even if that means we are not to be working as iOS application developers.

We are very satisfied with our application and how it turned out. The features and the design did not differ from our original planning in any way, if anything it exceeded our expectations of the product and our own abilities.