Final Report

Chan Chuan Sheng (27465888)

Agile software development practices

For our development of the Morse Code decoder application, we have adopted several Agile practices for both of the iterations. As a result, we have gained a substantial productivity boost compared to our previous assignments when those practices were not integrated in our workflow. Some of the Agile practices that we adopted in our development are sprint planning, progress tracking using a project management application and finally sprint review after we have completed each sprint/iteration.

Before we started working on the user stories given in the specification sheet and implementing them, we had a brief sprint planning session in which we split up some of the user stories into smaller tasks which are easier to be dealt with individually. By maintaining a list of sorted tasks based on their priority, we have a more accurate estimate of the size of a particular user story by stating the amount of work for each task. Breaking down into individual smaller parts also helps in collaboration between my partner and I. Each of us can usually take up a task, and work with minimal conflicts and just a bit of adjustment to get each pieces working with each other. In addition, we can also track any missing pieces of a particular user story and get a measure of our progress in a sprint. After we have completed an iteration, we would conduct a sprint review session which both of us would sit down and talk about our problems and lessons learned from the previous iteration in order to refine our workflow for the upcoming iteration. This particular Agile practice yielded a huge amount of benefits as a lot of misconducts have been found out and thus rectified in the next iteration. Despite our heavy workload and time constraints of our schedules, we managed to have a daily meetings in order to keep track of the status.

Throughout this semester of taking FIT3140, I have noticed a profound change in my personal software development practice. I enjoy the change and welcome it as one of the habits during the build-up of my software development career. Except those that are mentioned above, I have also adopted the practice of test-driven development and problem solving skill from a generic viewpoint as well as code modularization for reusability. Not to mention that I also picked up one of the most valuable and essential skills of any software developers nowadays - Github, the repository hosting platform we have been using since the first assignment, that unleashes the full potential of collaboration among the developers who work on the same project. Having equipped with a powerful revision control system and seamless Git integration, Github provides a

perfect solution for any form of group project despite the slightly steep learning curve at the beginning stage.

There are some Agile practices that we did not adopt in our sprints. For example, we have omitted the part of design and modelling which is required in most Agile software development practices. The reason we opted not to include UML design and modelling as one of the initial phases was that UML diagrams were not deemed necessary in our assignment as the the set of user stories given are straightforward and fairly simple to be implemented. While UML diagram is only suitable for a huge and complex system, it was a complete overkill for a Morse code decoder application.

Thinking back over my project, there is only a thing which I like to change. The way our group handled each individual tasks was to split by assigning a subset of them to a member and then the rest of them to another member based on the rating of the size and then we will stick to this until the end. By adopting this approach, we basically restrict ourselves from helping our partner if we finished the task we were assigned to. Even if we wanted to help, we had little to no knowledge of the progress and the implementation details. Hence, I would like to change it to a user story-based separation of tasks. Each of us should be assigned a task from a single user story and then proceed to the next task of the same user story until we finished implementing all of tasks. With this approach, each of us would know the implementation details of the user story and is able to help out our partner more easily.

Working in teams

The coordination between my partner and I mainly relied on the Trello board. We maintained a list of sprint backlog which kept all the split up tasks from the user stories along with their corresponding sizes and the person who was responsible for it. Once we were done with the task, we simply moved the card to another list called ‘Done’ which kept all the finished tasks. We found out that by using Trello, it actually allowed us to keep track of the progress of the project easier.

Throughout both of the sprints, we were constantly kept in touch via WhatsApp, an instant messaging platform. Before each meeting, one of us would try to contact the other and asked for an available timeslot and since we both share a similar timetable, it was quite easy for us to meet up. Usually when we had some doubts or enquiries, we would just discussed about it in WhatsApp. I personally encourage this form of communication as it allows us to exchange thoughts conveniently in real-time.

We allocated our tasks based on the size of each tasks. For example, if there are four medium-sized tasks and 6 small-sized tasks, we would divide them equally between two so each of us would have to work on two medium-sized tasks and three

small-sized tasks respectively regardless of which user story those tasks belong to. Although this particular task distribution approach was practiced to guarantee fairness, it is in fact flawed to a certain a extent. One of the reasons being the one which has already been discussed above - it minimises the potential of collaborations. If a member has some issues or troubles in his task, it is quite difficult for another member to help out as he does not have enough knowledge on the implementation. Besides, there might be some inaccurate estimation on the size of a task. For example, the task that both of us thought would be tough end up had a simple implementation.

I will continue using the abovementioned practices if I was programming in a small company. Perhaps there will be a more organized way of communication among the developers instead of WhatsApp, for example Slack. However, task tracking practices like Trello should be encouraged and used as it provides a lot of conveniences and benefits.

Design

Since our codebase is completely modularized, it should not be a difficult task for us to add into the design if the client ever demanded a variations support. The architecture of our application is a mixture of two different programming paradigms - functional and object-oriented. By employing functional programming practices, we are able to reduce the number of bugs in our codebase significantly.

Our architecture places most of the logics and algorithms on the server side. As a result, the client needs not to be a fast machine with a lot of memory to run the application. Our decoding algorithm is an online algorithm. It means that the conversion between motion signals and characters are done in real time. In details, we have an array which keeps track of all previous motion pattern. Once a character gap appears, the motions stored will be used to compare against the Morse code coding table for decoding process. Our client design is quite basic and would require further polishing before it's ready to be a production-grade UI. However, it does fulfil all of the requirements specified in the specification sheet.