**Agile software development practices**

With our project, we used a variety of agile practices. They include, planning, design, test-driven development, refactoring and continuous integration.

In the planning and design phase, we cut all the user stories up and stored them in a trello board for easy reference. This was very helpful because as a result, we were able to keep a track of time and be aware if we were able to meet the final deadline.

We also implemented test-driven development both formally as well as informally. Formally, we did black box testing and testing for the decoder function and recorded the results. We also performed a version of unit testing where the decoding method on the client side was tested on a separate file. Different inputs of Morse code were entered to see if they returned the results we expected. Informally, for almost every bit of new code we wrote, we did small tests on them. For instance, when implementing a push to firebase, we would check and see if that data appears on the Firebase database. Also, when doing the “SK” ending of transmission, we checked to see if the server really exits node js when the Morse code “SK” is sent off as a single word. This type of constant testing made it

Since our main priority is customer satisfaction, we know the customer is likely to change their mind sometime through development. For instance, our initial plan after assignment 4 was to send each Morse code word as an array, “[‘SSS’, ‘L’, ‘LLSS’]” and the client can decode them one after the other. However, the user stories said how it wants to determine if the current Morse code signal set by the server side is a long or a short, so we had to refactor our project to sending each Morse code signal by itself into client side. Therefore, another agile practice we used was refactoring.

We didn’t have daily meetings as it wasn’t necessary. The reason was due to continuous integration. Each time we made a big enough change to our code, we uploaded it to the GitHub repository with a commit message describing what was done in it. We also used Facebook to leave a more detailed description of the changes made if needed.

Thinking back on the project, we should have put more consideration into the complexity of each task. Most of these estimates were done quite accurately, but we under estimated the complexity for the final user story for iteration 2, where the client side is able to start and stop the transmission.

**Working in teams**

In our team, my partner was mainly in charge of planning, design and testing whereas mine was to program the solutions. However, we took this approach quite informally as we tried to help each other out. For instance, we would communicate over the phone about each other’s ideas and brainstorm until we reached the best solution. Also, when I am doing the coding, sometimes I may encounter a bug which he will then help me fix.

Most of our communication was done through phone calls and Facebook. This was the easiest approach for both of us as we found it more productive to do the work at our own homes.

Me and my partner did most coordination through Git-hub. When each of us made some notable change, we would usually message the other on Facebook and ask to check the repository to look at the made changes to see if they are happy with them.

When it comes to programming in a small company, I believe our approach will be adequate. We were both synchronised with the task at hand, along with what each of us were doing. However, we both preferred working from home. In my case, because I have a fast and reliable desktop computer that I find very comfortable to use. This could be problematic if we were required to work in a company that required us to be on site every day to make progress on a given project. However, this is only a minor setback as I could purchase a laptop with similar performance and be more mobile.

**Design**

When it came to writing the code, I took measures to make it run efficiently as well as trying to make it simple for someone else to understand. For instance, I used a separate function to decode each letter in the client side. When we were told to add more functionality, by decoding numbers and punctuations, I found this process to be quite easy as a result because all I really had to change was the decoding function and add some more Morse code entries and arrays for the decoder function to look up. Based on that, I would say that it is quite easy to add some more variation to the program.

Another aspect to notice is that we used separate files to make the system more organised. For instance, on the client side, the user interface design is stored in ‘index.html’ and the decoding algorithms on another code file, ‘client.js’. This way, a designer can easily change the user interface outlook without having to go through technical code. Vice versa of this applies when a developer is trying to change some back-end technical aspect on the client side. Also, lots of code commenting was also used to describe what each section of the code was responsible for.

When it comes to fixing bugs in the codebase, the level of difficulty will depend on how much experience the person doing so has with node.js, just as much as the design of the code. However, the fundamental complexity of this program is simple. I have included a readme file which contains a high-level description on how each part of the program works, along with lots of commenting on the code. This makes me believe it should be somewhat easy for other people to fix any errors that may arise, if they do.

When it comes to the visual design of the front-end user interface, which was done mostly by my partner, I am quite happy with it. The colours match quite well and it is very simple and easy to use. In fact, the UI looks so simple that even children can use it without much of a hassle.

Also, since the UI is stored in a different file, it is very easy to change the outer-look later on while keeping the same level of functionality.

**Summary**

Overall, this assignment was a great one as it allowed us to explore agile processes and get an idea about how important they really are in the industry. Agile also helped us stay organised and motivated through every step of the project. It also helped us deliver a product that was very close to what the user required by the end. This project also provided me with teamworking experience, which is something I find quite challenging and am willing to improve on.