**Individual Report**

**Introduction:**

Morse code app that we made checks with the dictionary (encoded in our code) and then well any long or short motion is detected by the PIR motion sensor, if the motions matches with the dictionary, then a word is printed in the app. Throughout the production of the app we ran through issues when we tried to run, sometimes the BeagleBone board (BBB) was not responding or when we ran through a series of tests, to check for bugs. More in details will be discussed in the “Content” part of the report. Our group consisted of Turki Baghlaf and yours truly, Suvashish Chakraborty. We followed the “Agile methodologies” throughout the development of the assignment, dividing up our work, planning the next step, meeting to discuss any issues, noting any amendments, planning the next step, running a series of tests to check for bugs, and continuously improving and optimizing our work.

**Content:**

* *Planning Game:*

We planned to tackle the assignment code first, as it was the crucial part of the assignment. Turki took responsibility for the codes, while I made the documentation and report. We met to plan how we would allocate our resources for the assignment, brainstorming ideas and noting if anything significant came along. Whenever a part of the code was done, we ran a few tests to check if the part responded well, if the response was as expected, we moved on to the next part of the code. Every meeting we discussed issues regarding the assignment. We followed “*Lean and Kanban Software development*” throughout the construction of the app. Lean methodology eliminates waste through such practices as selecting only the truly valuable features for a system, prioritizing those selected, and delivering them in small batches. It emphasizes the speed and efficiency of development workflow, and relies on rapid and reliable feedback between programmers and customers. Lean uses the idea of work product being “pulled” via customer request. It focuses decision-making authority and ability on individuals and small teams, and is more efficient than hierarchical flow of control. The method also concentrates on the efficiency of the use of team resources, trying to ensure that everyone is productive as much of the time as possible. It concentrates on concurrent work and the fewest possible intra-team workflow dependencies. The method also strongly recommends that automated unit tests be written at the same time the code is written. We acted directly to the mistakes that were observed, thus eliminating waste. We tried to assist each other and searched the internet for any information regarding the app development, while constantly improving upon our app, thus amplifying our learning, deciding as fast as possible on any changes we wanted to make on the app, and building integrity.

* *Design, Development, and Planning:*

Since agile approach follows a cyclic iteration of work, we check through every mode of development. Making test cases for user stories given and then moving on to inspection. For each of the three test cases we tabulate the results and resolve the test cases. The benefit of using the agile approach is we can backtrack to the situation we faced our issue, so that we can resolve it. The SCRUM meeting that we conducted from time to time, are mostly done through social media, where we discuss and search for solutions on the internet.

* *What we did not adopt:*
* Sharing the codebase between all or most programmers
* A single coding standard to which all programmers adhere,
* A common “war-room” style work area.

The first one was not adopted since we were already a group of two, so most of the coding work was done by one of us, and the rest were done by me. Moreover, we did not disclose our code or what we were doing to anyone, except to the faculty, so we were not sharing any information. We planned to use JavaScript(JS) and Python in our coding, since we worked with both socket.io and Firebase Web application for the development of the application. Therefore, we did not just simply follow one of the coding standard. Turki and I did not plan our group meeting in a singular place, we would switch rooms inside Monash University to test certain part of the assignment, as the PIR motion sensor was very sensitive towards even the slightest motion, and moreover, the internet connection problem was a big issue when we wanted to test out Firebase, as it heavily depended on the internet connectivity.

* *Anything that we wanted to do differently:*

We tried to make most of our time, so we proceeded step by step. Testing for any errors that we made. We acted directly to the mistakes that were observed, thus eliminating waste. Turki and I endeavoured to avail each other and probed the cyber world for any information regarding the app development, while perpetually amending upon our app, thus amplifying our cognition, deciding as expeditious as possible on any transmutations we wanted to make on the app, and building integrity. So, I think we only wanted to make the user interface for the app better than it looks like.

**Working with the team:**

Turki is productive, does his work in time, and communicates immediately if he faced any problems or any irregularities in the code or the documentation. We mostly did our communication through the internet for any small problems, but if it was a major issue, we organized meetings in Café’s or the university library. We orchestrated to tackle the assignment code initially, as it was the crucial part of the assignment. Turki took responsibility for the codes, while I made the documentation and report. We met to orchestrate how we would allocate our resources for the assignment, brainstorming conceptions and noting if anything paramount came along. Whenever a component of the code was done, we ran a few tests to check if the component responded well, if the replication was as expected, we moved on to the next part of the code. Every meeting we discussed issues regarding the assignment. Since, this is a group of two people, the risk of having scope creep is decreased, however in a large group the problem is forever present, unless the scope is dealt with immediately. Since, we noted down ay issues we came across and faced them head on, in a group, if cooperation is not present, the issue will not be resolved so easily. So, we need to have SCRUM meetings, on a weekly basis, to discuss certain issues regarding the problems being faced by the team and deal with it immediately.

**Dealing with modifications:**

It depends on how much the client wants the code to be changed, as there several issues that we need to think about when we are amending the codes to meet the client’s needs. Fixing the bugs in the modifications will be top priority, however we will not be taking regular orders to make changes, as we will be constantly updating the application. Once the company decides to make major amends or report major bugs in the application on the, only then can we make a move. Changes take at least a month or two months. We are not happy with the visual of the app, as it is a basic look to the app, with no fancy background or buttons, just a simple web interface. Adding more modifications is easier, and moreover, the usability will be improved with each update, as it will retain the basic touch but with more visuals and more add-ons on the application. The application is safe to use for all ages, and even the adults can use it to teach the children how to decode Morse code.

**Conclusion:**

The application was made to cater to the older audiences, who might need a Morse code interpreter while they are travelling or hiking. However, it will have more add-ons to add more flavour to the app. The Morse Code interpreter was made with extreme diligence to details and updated with the latest Morse Code dictionary.