* How do I interpret user needs and implement them into a program? How does creating “user stories” help with this?

The use of user stories is incredibly critical, but it’s important to note that the stories must be properly written to get the maximum value out of them. If we have poorly written user stories or stories with minimal details it leaves too much up to the development team to interpret what features the story is trying to represent. By having well-formatted and clear user stories, the development team has definitions of features that are outlined by real-world examples of how the final product will be used.

* How do I approach developing programs? What agile processes do I hope to incorporate into my future development work?

Throughout the course, I have felt that the combination of both pair programming and the sprints to be the most beneficial ideas to take with me into the workforce. In prior technical jobs I’ve held, working with other people directly during projects helps keep the solution more open and with a greater focus on overall efficiency. Having input from multiple individuals helps the project remain more objective with its means of completing a task which I believe helps the final feature being developed.

* What does it mean to be a good team member in software development?

Being a good team member I believe relies heavily on maintaining an open mind and positively accepting input from other team members. Allowing yourself to remain open-minded to solutions that you don’t come up with will only help keep the project moving forward with maximum objectivity. In an ideal workplace, the whole team would have like-minded perspectives, this however is almost guaranteed to never be the case.