Scrummage is the time in rugby a team takes to reset and adapt to their opposition. It became apparent that software development had similar needs and Scrum was born. The methodology is used with agile to allow businesses to keep up with a rapidly changing environment. (Cobb, 2015).

Each member of the Scrum Team serves a crucial role in the success of a project, while allowing the project to remain agile. Those roles are the Scrum Master, Product Owner, Developer, and Tester.

The Scrum Master’s primary job is a facilitator. Talking with developer friends, I learned that many times the Scrum Master also serves as a developer and most of those teams call them a lead rather than a Scrum Master.

The Scrum Master will lead stand ups and scrum reviews. They help direct resources to needed areas of a project as developers meet blocks. They collaborate closely with the product owner and strive to understand the product more fully to help others when the Product owner is not available.

In our case, the Scrum Master managed the immediate needs of the teams. The example case in class managed absences, tardies, and directed developers to pair to overcome blocks.

The Product Owner was a name I initially didn’t understand. But as I studied, it became clear that they didn’t own the product, but more they tried to understand it the way the client, the actual owner, saw it.

In week 5 discussions, Tyler Corliss posted about a project that he oversaw. In that case, he worked to understand what the department head and the end users expected. He then worked to reconcile the vision and needs to deliver a product that everyone was happy with. That struck me as very close to what a Product Owner should do to deliver a superior product.

The Developer produces the software that is ordered, as in our travel project. They, in combination with the team leads, help establish what is and isn’t possible by voicing what they can and cannot do in standups and other scrum meetings. They may decide on some things based on what they understand. This can sometimes lead to sudden pivots as the Product Owner presents workable software to the client.

The Tester designs and tests the software to make sure it meets the needs and passes real-world tests. I liked the example in the reading about the plane that was ready for testing and they allowed people to come in and see it. A bunch of children started pounding on the buttons, causing the software to crash so badly they had trouble getting it restarted. (Cobb, 2015)

Testing involves more than just testing the software pieces, but testing them all together. The developer can do a lot of this and should before sending the software to be tested, but an extra set of eyes on a problem can only help.

Another important part of the Tester’s job is to let the developers know where they find any problems to help keep future problems from arising.

The Manifesto for Agile Software Development has led to the creation of a few different models that are similar in a lot of ways. The basics of Agile are:

Individuals and interactions over processes and tools.  
Working software over comprehensive documentation.  
Customer collaboration over contract negotiation.  
Responding to change over following a plan (Beck, et al., 2001).

These are covered with Scrum. Its approach focuses on delivering useable software packages that can be adapted and changed throughout the development cycle. The Product Owner is constantly working to get feedback and deliver a working product that fits the customers changing needs such as in the case I noted earlier where after seeing the product the client asked for the site to be changed from a list to a slideshow presentation.

In a waterfall type development, this may not have been something that was possible as the customer would be unaware of what the product looked like until we delivered the final software.

With scrum, a product is complete when it passes all the tests. Because of the client’s ability to change and adapt their needs at any time. A project may never be completely done, but we should deliver a usable product at the end of each sprint as software that passes a user story test.

In last week’s discussion, I played the role of Scrum Master. I clearly outlined my roles and responsibilities and cited the direction of the team, but did not get to put them into practice. We discussed where team members could look for answers and others provided excellent input on what needed to be done.

Stephanie suggested we create an Agile Team Charter which would have been a great direction to go had we more time. The charter would have highlighted the goals and success criteria and helped focus the team on getting the items done.

Throughout the various discussions, Kaylee and I both encouraged the other team members to be open and provide feedback. This is important to the success of an agile based system. In this week’s reading, many of the problems the company had were communication based (Cobb, 2015). Without communication, the agile method is dead in the water.

Our success came as we worked as a team. Using the discussion board as a communication platform was useful; In our case, we were discussing our individual roles, but we could easily shift those discussions to a user story that needed to be addressed. The other tools, such as a sprint planner or the user test template were useful in providing easy to look at consistent information.

Tools like these help organize and manage expectations. They put information into easy-to-read visualizations that help make large projects more manageable.

For the SNHU Travel project, the agile method seemed appropriate. We could have put more time and effort into the initial planning that could have provided a better idea of what the client wanted. But overall, it was the right choice. Without the Agile method, we would not have been able to pivot to the client’s needs without difficulty. That’s not to say that we didn’t have any problems.

Challenges included not having a good idea of overall design, not knowing what database the client was working with, and changes to the top offerings that they would be making. With as many changes and as often as they seemed to happen, it might feel like the project would never be done.

The same things that are problematic also give agile strength. The reality of any environment is that it is constantly changing. Being able to adapt to those changes gives businesses and individuals the ability to keep up.

Overall, a scrum-agile approach also helps prevent the deadline death march by allowing the team to come up with realistic expectations. It keeps software development focused on the end user. And importantly, it keeps the customer involved and allows them to respond to an ever-changing environment.

# References

Beck, K., Beedle, M., Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., . . . Thomas, D. (2001). *Manifesto for Agile Software Development*. Retrieved from https://agilemanifesto.org/iso/en/manifesto.html

Cobb, C. G. (2015). *The Project Manager’s Guide to Mastering Agile : Principles and Practices for an Adaptive Approach.* Wiley.