A Brief History of Agile and what it attempted to solve in software development

Oh Agile, the darling of the software industry. If you know any history of software development you know that Waterfall was the method of choice for a long time where we just moved in a linear way through development. This made sense when we were building complex enterprise applications that took years and millions of line of code. Believe or not these behemoth software packages are still in existence and your information is being stored on it right now!

Interestingly, as web technologies have become more sophisticated, we can do some pretty amazing things, we get further and further away from the backend that serves it. Agile came about to solve two problems that developers were having, 1) the higher-ups kept changing the scope after a lot of work had been done, and 2) no one was paying attention to the end user. When software was first created, it was for engineers, by engineers, and they were pretty good at making stuff that worked for them. Once the rest of us got computers, the complex stuff engineers made for themselves, all of a sudden, didn’t work for everybody. Interestingly the concept of who the user is for Agile is not exactly what we would define as the user in UX. The user in Agile tends to be a bit of an abstract concept and many decisions can still be based on assumptions. When a UX team is involved, they bring in real world knowledge of the users and users stories reflect that.

Software development methodologies such as waterfall didn’t address user issues or the fact that stakeholders would want to change course faster than engineers could keep up. Agile came in to solve this by demanding an organization put the user first and allow the engineers to work in smaller intervals. The people who started Agile didn’t realize that there already was something that could take care of the users, and that is of course User Centered Design.

Many engineers have never seen a designer in the wild or worked with one so they didn’t know how we could be of benefit to each other. By the time UX met up with Agile, we noticed a few issues. It’s kind of like trying to blend two families, like the sitcom, *The Brady Bunch*, where each family came with their own culture and had to figure out how to live together with a new set of people in the mix. Agile and UX continue to reevaluate how to coexist and also how to collaborate. Many methods have been developed to address this and make sure that the two can work together. As you learn more about the prototyping process you will likely see where the two philosophies between Agile and UX align very nicely and a few where they have some major conflict.

Agile & UX how do they work together and how does that affect how we prototype/iterate design?

Many UX teams struggle to integrate with an Agile team. If you think of Agile like a train you can imagine it is really hard to stop it mid course. You have to wait until they get to a scheduled station before you can get on. For UX we like to be more like a bus, making frequent stops, some can be added or taken away without much notice, routes can vary based on traffic and we can gather new passengers along the way. Getting a train and a bus to line up at the same station at the same time is the major challenge. What this turns into for prototype testing is that we have to get as much done as possible ahead of the next train stop in order to make sure our changes make it on the train. There are many resources out there on how to integrate and I've included some in this week's readings. As you work with Agile teams, you will develop your own tricks and techniques that are unique to each company you work with.

What is Lean UX? Is it really a different thing than Agile UX?

If you do a quick Google search on Lean UX and Agile UX, you may find some confusion and to be honest, I easily get confused too. I think in UX we really love the design process and are constantly trying to make it better. We use new terms to describe old things all the time. Being user centered is not enough of a mandate anymore, we need to also be more efficient in how we approach our work because there is never more time or budget available than what we get. In UX we still have a lot to prove to businesses that this is a valuable way to build products. Many of the descriptions of Lean UX sound a lot like Agile and that is probably on purpose: to both better integrate into the team and to get approval from the business who has already invested in Agile.

One area where Agile has really influenced UX is with documentation, and the philosophy that less documentation is better. This is tricky because as designers we really need to hold the entire product vision together and in doing so, we need to have documentation that effectively communicates what that vision is. If we take the idea of less documentation too literally, we have few artifacts to fall on in order to deal with miscommunication.

In this course we will learn how to be user centered, Lean and Agile in our approach.

We will move quickly not letting ourselves get caught up in perfectionism. Remember that train is coming! This doesn't mean we sacrifice quality product design, but it means we do things efficiently and focus on what matters.