

But Agile says....



As an experienced Scrum Master I hear this phrase a lot. Sometimes it is from people who don't really understand Agility and think it means 'do what you like'. More often it is said by people who do have some experience with Agile methods but are just quoting part of a framework or a concept in order to follow a certain agenda, rightly or wrongly.

I have heard in the past a sort of Agile one-upmanship occasionally going on:

"The important thing in Agile is to get rid of waste."

"Scrum events are not really waste. They're part of the Agile framework."

"Isn't it part of the Agile framework to self-organise? If we decide to get rid of them then that's still Agile."

Etc. etc. These sometimes end with:

"Well the Agile coach said..." which doesn't really help as we know that coaches often have different opinions (as we are, believe it or not, human like you.)

So what is Agility? Well maybe it would be helpful just to sum up some of the major players and try to get a simple definition. Let's start with Lean.

The basis of Lean methodology is efficiency. Understanding the value of a thing and removing anything that doesn't contribute to creating that value. This keeps the flow of work going. Lean also introduces the idea of finishing a piece of work before starting a new piece, only pulling new items into a work stream when others are finished. It also insists on a way of self-examining in order to improve the system.

Now a shallow dive into Scrum.

Scrum has a set of values that it would have its practitioners practice and probably the most apt of these is focus. Because Scrum is about deciding exactly what to build in a short period and then focussing on people working together in order to make it, instead of using specification documents. A series of feedback mechanisms contribute to both the creation being of a high quality and enabling the team members to improve the manner in which they work.

Scrum and Extreme programming were developed at about the same time. What is XP all about?

Extreme Programming has a more granular focus in that it consists of a set of engineering practices that both 'trim the fat' of the process so that developers can get on with making things but introducing lightweight checks and balances in order to make sure that the thing concerned is still of a high quality. Substituting specification for simple 'user stories' and then using relative estimating to let the team understand how many things they can make in a period simplifies the process. Quality is improved by using practices such as pair programming and test driven development. This is supported by team ownership.

Lean, XP and Scrum were instrumental in the formation of the Agile Manifesto (it's a common misconception that this came first...), a kind of one stop shop for Agile concepts.

It's easy to just repeat verbatim the four tents of the manifesto (and the twelve associated principles) but it's possibly more useful to read between the lines. For instance it's clear that you should try to defer to direct communication both inside the team and outside. Because this is still the form of interaction that is most likely to give accurate information. But at the same time it questions how much information you really need in order to create 'working software'. Also the manifesto wants one to question the value of information and the thing you are creating itself. Making sure that when change happens then it is right to ask whether a piece of work should be continued to be progressed.

Summarising these concepts then we can see:

- Lean – Build the right things at the right time
- Scrum – Build small increments of things in a small amount of time.
- XP – Work together as a team to build and design code. Using automated testing and test driven development catches problems early and improves quality.
- Agile Manifesto – Maximise communication and minimise extra work.

So having reviewed some of the main methods and ideas are we closer to finding a simple answer to the question 'What is Agile?'. If we had to sum it up in an elevator pitch style, can we? Let's have a go:

**Agile is working in such a way which minimises uncertainty and mitigates risk.**

Using this as a definition means that if you want to check that you are using Agility then both of these conditions need to be satisfied. For instance you can minimise uncertainty by spending months on upfront design. But by doing that you are increasing risk as both the technology, market for the product and client intention may change in that time.

I've generalised a lot regarding the different methods mentioned above (or perhaps used 'lightweight thinking' ...take your pick) but what is useful regarding this simple definition is that it gives a quick and easy way to think about whether a process really is Agile or not.

And they say loaded dice are a bad thing...