

Agile – What it Means

Let's start with the Agile Manifesto as proposed in 2001.

Individuals and Interactions over Processes and Tools

Working Software over Comprehensive Documentation

Customer Collaboration Over Contract Negotiations

Responding to Change over Following a Plan

As the Manifesto states, it is not that the things on the right are not important at all, it's just that the things on the left are more important and therefore need to be emphasized. But just what do those four sentences mean? What is really important in an Agile environment?

Continuous Interactions between Different Groups

Every project consists of multiple groups; users, developers, designers, managers, etc.

In a waterfall project, those groups get together at specific times for specific durations, generally through formal meetings. Once those times and durations are over, the groups then work separately for much longer periods of time.

But Agile sees the connections between the groups being more of a continuity instead of nodes. Agile sees connections as an on going conversation that continues at a pretty constant level over the life of the project.

Waterfall is a bowling or golf league. Agile is like living in a commune.

Having Faith in the Team Members

Faith is a difficult thing to define and quantify.

Very few waterfall project proponents would say that they ‘don’t have faith’ in their team members. But there is a difference in the way Agile looks at that faith.

The Waterfall methodology has faith but – it also has a series of documents that need to be created prior to any actual work occurring that makes sure people live up to what the project needs. The underlying feeling is that without those documents, nothing good is going to develop.

Agile, on the other hand, takes as one of it’s core tenets that team members will be responsible and quality oriented enough to develop the thoughts behind these documents as they discover the functionality they are building.

In the end, Agile knows that a document outlining what will be done and agreed upon by several groups is not a requirement for starting to work on a project – the work and the agreement on it will be done in parallel thanks to the continuous interaction between the different groups that we described above.

Finally, and this is very important, Agile accepts the fact that functionality recipients will only respond to real, working things, not documents.

The fallacy that Waterfall embraces is assuming that everyone will read and really, really think about the written documents describing that functionality.

Sadly, everyone today is double booked and doesn’t have time to really carefully read and digest written documents. To be honest, my feeling is that even if people were just sitting around, they wouldn’t look at a written document all that carefully. Most people just aren’t built that way. People really won’t start paying attention until they have working software to use.

Agile realizes this and so it puts the focus on getting this working software into the end users hands to get immediate and solid feedback on whether this is really what they wanted. Doing this in iterations reduces the time required to do this and so reduces the potential lost time because what was developed is not what people really want.

Agile is not user stories or epics or Scrum or test driven development or any of the other hundred things that are bundled with it. It is four very basic principles that say, in essence, people who are talking directly and continuously to each other, trump methodologies and procedures. Very simple, actually. And very effective.

Want more info on Agile? Or Waterfall? Or how **SCS** can help your company with it's project management issues? [Cool](#).