

Principles behind the Agile Manifesto

The [Agile Manifesto](#) is backed or supported by [12 principles](#) that describe the implementation approach. As mentioned in a related post, Agile is often blown up to a far more complex idea, with a misguided thinking that to be agile it means certain tools, specific techniques or other miscellaneous trivia. At its heart, the idea is simple and compelling. Breaking down the principles makes that clear, as you can see the roots in other approaches rolled up into this manifesto and related principles.

We follow these principles: (**Emphasis** added by me)

1. Our highest priority is to **satisfy the customer through early and continuous delivery** of valuable software.
2. **Welcome changing requirements**, even late in development. Agile processes harness change for the customer's competitive advantage.
3. **Deliver working software frequently**, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must **work together** daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and **trust them to get the job done**.
6. The most efficient and effective method of conveying information to and within a development team is **face-to-face conversation**.
7. **Working software** is the primary measure of progress.
8. Agile processes **promote sustainable development**. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. **Continuous attention** to technical excellence and good design **enhances agility**.
10. Simplicity—the art of **maximizing the amount of work not done**—is essential.
11. The best architectures, requirements, and designs **emerge from self-organizing teams**.
12. At regular intervals, the team **reflects** on how to become more effective, then tunes and **adjusts** its behavior accordingly.

Agile takes multiple forms, and a couple of the most common are the foundations of many more. Kanban is a more sequence driven form, and Scrum is probably the most common team & sprint based approach.

There are a variety of tools used to aid in the management of agile projects, but the clear leader is the Atlassian tool called Jira. Of course, Excel is a fast follower as well in the spirit of keeping it simple!

I will post more on this topic, as I am in the middle of helping to drive an Agile Transformation in our enterprise R&D area. We have baselined our current approach and now my team is asking “why not agile” as opposed to “why agile” or assuming waterfall as the delivery process.