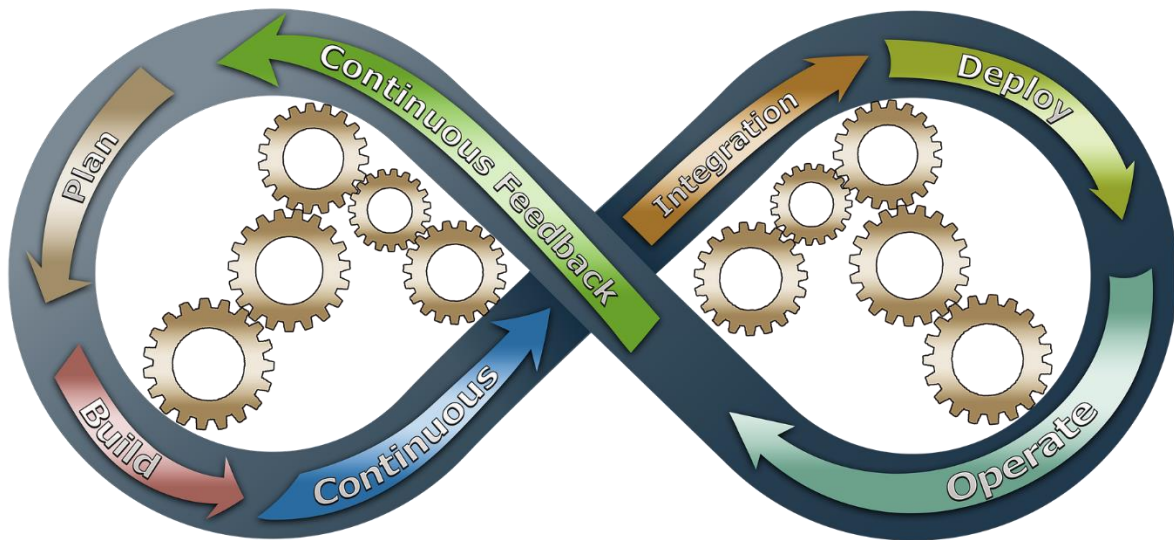


What is Agile and How Does it Differ from Waterfall?



In the 1990s, it became evident that software development was unlike other engineering development processes. Up until that point, most engineering development was done in a more linear mode known overall as the Waterfall project delivery process.

The Waterfall process didn't move fast enough to ensure that customer's software demands and requirements were being met. This is because the completed software generally was not available to the customers until at least three years had passed. The software needs may have changed by then. Something had to be done!

Agile came about due to a meeting of 17 individuals in Utah in early February, 2001. These individuals were "representatives from Extreme Programming, SCRUM, DSDM (Dynamic System Development Method), Adaptive Software Development (ASD), Crystal, Feature-Driven Development, Pragmatic Programming, and others sympathetic to the need for an alternative to

documentation driven, heavyweight software development processes convened.”¹

A Manifesto for Agile Software Development was developed and signed by each of these representatives and the Agile methodology was born. The values of the *Manifesto for Agile Software Development* are listed below.

The values on the left side of each statement are more important than the values on the right in Agile methodology. Remember that when you deliver software and other products using the Waterfall methodology, everything is front-loaded, decided upon, and signed off on prior to development beginning. That’s not the case with Agile.

“Values

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan

Principles

1. Customer satisfaction through early and continuous software delivery
2. Accommodate changing requirements throughout the development process
3. Frequent delivery of working software
4. Collaboration between the business stakeholders and developers throughout the project
5. Support, trust, and motivate the people involved
6. Enable face-to-face interactions
7. Working software is the primary measure of progress
8. Agile processes to support a consistent development pace
9. Attention to technical detail and design enhances agility
10. Simplicity
11. Self-organizing teams encourage great architectures, requirements, and designs
12. Regular reflections on how to become more effective

Those who apply any type of Agile methodology adhere to these values and principles. The manifesto offers a good overview of what is expected when it comes to the Agile development life cycle practices.”²

Based on my own work experience as a cross-functional project manager in a large multinational corporation, I would say there are a lot of advantages to the Agile methodology. These advantages exist not only for the software development community but also for any other projects and deliverables that are iterative and incremental.

For instance, my organization’s deliverables were translation, documentation and training. Agile project management made the software translation process much more manageable and effective.

- Instead of getting ALL of the software at the end of the development cycle and then having to translate it into the required number of languages just before a product Launch, as we had done using the Waterfall process, we would now receive software iterations every few weeks. Each software iteration would be translated into the required languages when complete.
- The Software Development Team and the Translation Project Team would meet on a regular basis (sometimes daily) in the meantime to ensure both teams were synchronized with one another and to answer one another’s questions.
- If things fell behind on either side, the two groups would come up with a remediation plan and present it to the Product Delivery Team (PDT) to show their plans to get back on track to achieve a multinational Launch on schedule.
 - If this had happened when we were using the Waterfall process, it would have happened closer to Launch and the responsibility for the remediation plan would have been on the Translation team.
 - Under the Waterfall process, the Translation team would be blamed and held accountable for a missed Launch, even if it was a glitch in the software or a shortfall in the ability of the graphical user interface (GUI) that caused issues displaying some of the longer translated languages.
 - These difficulties would not happen when using Agile methodology.

Similarly, the documentation and training team was also working with the software development team in an incremental and iterative mode using Agile decision making.

- The two teams would be working in concert with one another in the development of the various product features' software, documentation and training.
- The two teams would frequently meet prior to the handoff of the tested software to the documentation and training team.
- Questions were answered and decisions were made in a real-time mode versus weeks or months down the road when the completed software was delivered in its entirety.
- Contingency and remediation plans could also be developed by the two teams in real time if something came up along the way that might jeopardize the Launch schedule or the budget.
- Those contingency and remediation plans could then be reviewed with the PDT and decisions made as soon as possible, before the effects of the issues became insurmountable.

The Agile Values and Principles are the most important things for the teams to remember and follow. The teams' practices to achieve those values and principles may change over time, from Scrum to Kanban or others. It's up to the teams to decide what's the most effective to use.

About the Author

Sandra Glanton is the owner and managing consultant of Projects Accomplished! She spent 23 years working in various phases of product development at a local multinational corporation. She also was a cross-services project manager for 11 years in an organization that specialized in documentation and translation services. She can be reached at sglanton34@gmail.com or (585) 230-0649.

¹ Highsmith, Jim, Agile Alliance. 2001. *Manifesto for Agile Software Development, History: The Agile Manifesto*, <http://agilemanifesto.org/history.html>

² Muslihat, Dinnie, Zenkit blog's content extraordinaire and productivity pundit. 2018. *Agile Methodology: An Overview*, <https://zenkit.com/en/blog/agile-methodology-an-overview/>