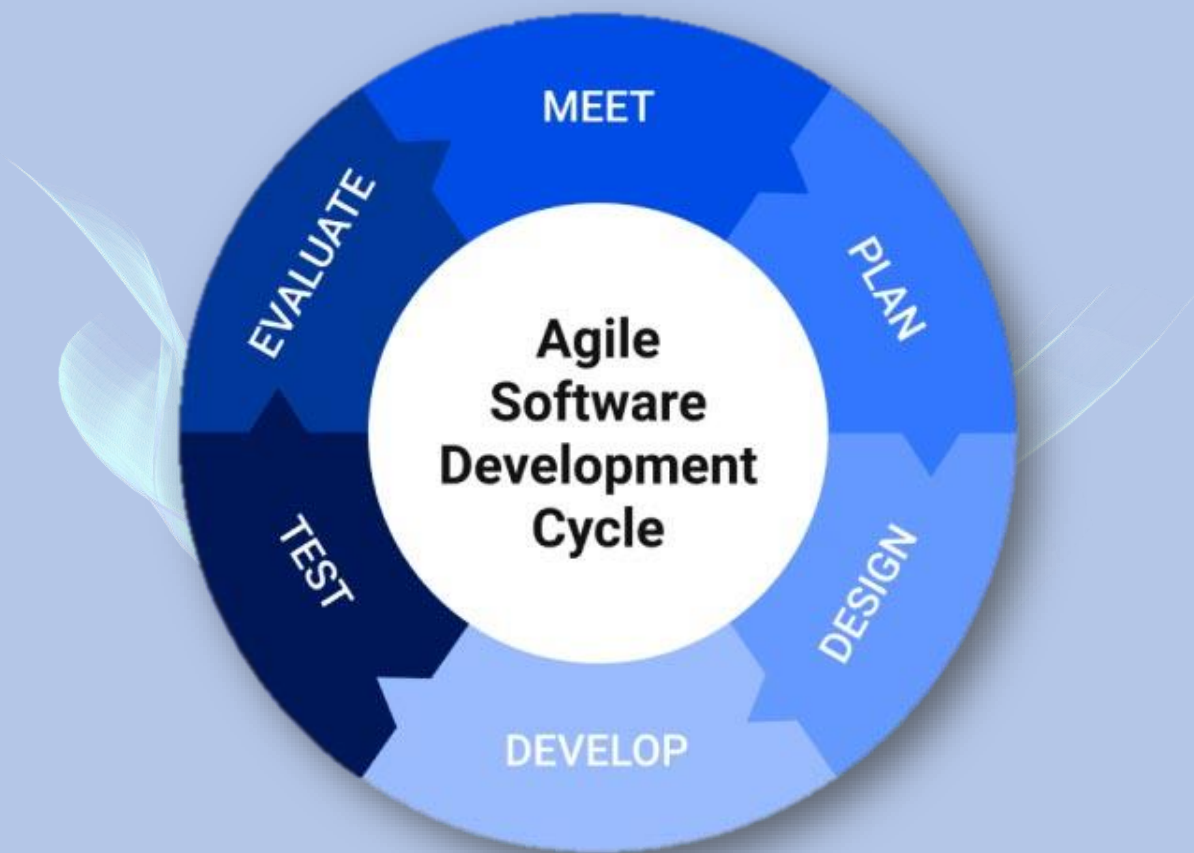# The Art of Agile

Investigating the concepts of Agile development and methodologies

Author: Maurice Schippers
Class: S3-DB03
Date published: 30th of November 2022
Version: 2.0

Contents

1. The foundation

Agile software development all started with a certain mindset and finding an alternative way to the traditional waterfall-approach. Taking 'Agile' as the representative word of choice, it can be found in the core values such as "delivering fast, details later", flexibility and quickly responding to changes.

The twelve Agile principles, as according to the Agile Alliance:

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 behaviour accordingly.

(Agile Alliance, n.d.)

To further establish a strong foundation, the 'Agile Manifesto' was created, based upon these twelve principles. The Agile Manifesto has four important 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

(Agile Alliance, 2022)

Software development teams then took certain principles to focus on, resulting in the many agile methods we have today.

2. Agile methods

According to the Dutch consultant and author of several Agile books Henny Portman, there are currently over 40 different agile methodologies. (Portman, n.d.) With some frameworks even forming hybrids with other frameworks like ScrumBan, the list of methodologies quickly grows. While Scrum, XP, Kanban, Crystal and Lean are arguably some of the most popular frameworks, depending on whom you ask, others may be included as well. Here is a handful of the most common frameworks:

1. Scrum
2. Kanban
3. Extreme Programming (XP) (Popularized Agile)
4. Crystal
5. Feature Driven Development (FDD)
6. Lean Software Development
7. Scaled Agile Framework (SAFe)

3. How to Scrum

Scrum is the most commonly used framework in Agile. It is widely studied and adopted by sectors even beyond software development. The most important concepts are highlighted. See figure 1 for a quick overview of these concepts.

In Scrum, teams work in 'sprints': a time period of two to four weeks. During this time period, there are a number of planned meetings, starting with the **Sprint Planning** on the very first day of the sprint. The Sprint Planning involves the Scrum team**, Scrum master** and the **Product Owner**. The Product Owner picks the highest priority items from the **Product Backlog**, and the Scrum team then picks the items they are confident in being able to finish within the sprint. This is how the **Sprint Backlog** is created.

After the first day of the sprint, the Scrum team will hold **Daily Scrum** meetings where they share what they finished the day before, what they will be working on today and any issues that they have encountered and are blocking them from completing the task. The latter is where the Scrum master comes in: their task is to make sure everyone in the development team is working according to the Scrum framework and to identify possible obstacles found by the team and remove them.

On the last day of the sprint, the Scrum team will hold two separate meetings: **Sprint Review** and the **Sprint Retrospective**. The Sprint Review involves the Product Owner and possibly other stakeholders. During this meeting the development team demonstrates everything they completed during the sprint. Completed work includes work that is 100% finished and can be used by customers and does not need to be worked on anymore. This meeting is used to receive (early) feedback from the Product Owner and the other stakeholders, in order to measure whether the product (still) meets the requirements.

After the Sprint Review, the Scrum team holds the Sprint Retrospective: A meeting to reflect and adapt in order to improve the process for the next sprint. This could mean taking out activities or adding activities and so on. There are several ways of conducting the retrospective.[1] Finally, a sprint truly ends with creating an **Increment;** an Increment consists of all the work completed in the sprint

---

[1] See https://miro.com/guides/retrospectives/ideas-games for more retrospective idea's.

and all the previous work completed in other sprints. A new Increment is created at the end of each sprint.

Figure 1

Scrum Process.

Source: (PM-Partners, 2021)

4. Frameworks different from Scrum

4.1 Crystal

According to senior software developer Susan May, in contrast to Scrum, Crystal is a more flexible methodology that can be tweaked based on the project and the team members. Since team members are in charge of making decisions regarding the methodology, it might be too challenging for Agile starters to begin with. That same flexibility brings along some experience. Additionally, Scrum has a Product Owner where Crystal sees the end-user as its Product Owner. Other small differences are that Crystal typically only has one week per iteration, whereas Scrum has at least two weeks. Bottomline is that while Scrum is meant to implemented in its entirety, Crystal is there for seasoned developers to take what they need from the framework and use what they feel is necessary. (May, 2021)

4.2 Lean

Lean has five main principles:

1. Identify Value
2. Map the value Stream
3. Create flow
4. Establish pull
5. Seek perfection

According to cloud-based project management platform Monday.com, Lean Software Development is often seen as a more principle-based methodology than an actual framework. One of its key differences with Scrum is that it is that there are no iterations, just <u>continuous improvement</u>. Furthermore, Scrum is meant for teams of about 10 people. Lean is sometimes used for entire companies. The main power of Lean is focussed on reducing waste and optimizing a process, in contrast to Scrum, where innovations and rapid change of a new product are the focus. (monday.com, 2022)

4.3 Kanban

Kanban has a lot of similarities to Lean and is all about visualizing teamwork. One of its most prominent features is the "Kanban board", where cards with items on them are moved from column to column (to-do, in progress, complete), depending on the stage of the development process. Using the Kanban board, everyone can be involved in monitoring the process and measure the time of completing certain tasks. By doing so, teams can learn from the results and improve from there on. Kanban strives to limit the number of items that are a work in progress, this is to allow continuous delivery to be possible. Finally, a major difference between Kanban and Scrum is that there are no roles involved in this methodology.

When software developer teams expressed their need for the structure Scrum has in combination with the flexibility and visualization of Kanban, ScrumBan was born[2].

4.4 XP

5. A day in the life of a Scrum professional

For almost a year now, I have been subscribed to the weekly newsletter from Mike Cohn, one of the founders of the Scrum Alliance. He shares his experiences on the workplace in these newsletters, giving a good look into Scrum in practice. Flexibility of adapting on the spot; like the scope (never delivering late) and the continuous learning in Agile. Pair programming with remote teams. Daily stand up, actually standing. TODO

6. Agile management tool recommendation

I have experimented with several tools like GitLab and Trello. My personal recommendation is therefore Jira. Changing is easy and the overview is clear. Has all the tools required for most Agile frameworks. TODO

7. References

Agile Alliance. (2022). *The Agile Manifesto*. Retrieved from Agile Alliance:
https://www.agilealliance.org/agile101/the-agile-manifesto/

Agile Alliance. (n.d.). *The 12 Principles of Agile*. Retrieved from Agile Alliance:
https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/

---

[2] See https://www.planview.com/resources/guide/what-is-scrum/lkdc-what-is-scrumban/ for more information about ScrumBan.

May, S. (2021, October 5). *Scrum, Crystal, Extreme Programming: Differences & Similarities*. Retrieved from Zeolearn: https://www.zeolearn.com/magazine/scrum-crystal-extreme-programming-differences-and-similarities

monday.com. (2022, February 1). *Lean vs. Scrum: what's the difference in 2022?* Retrieved from monday.com: https://monday.com/blog/project-management/lean-vs-scrum/

PM-Partners. (2021, June 23). *The Agile Journey: A Scrum overview*. Retrieved from PM-Partners: https://www.pm-partners.com.au/the-agile-journey-a-scrum-overview/

Portman, H. (n.d.). *A Project Manager's Guide to 42 Agile Methodologies*. Retrieved from The Digital Project Manager: https://thedigitalprojectmanager.com/agile-methodologies/