**Agile Methodology**

Agile is a popular technology that allows you to build and respond to changes. It is a collection of several principles that are used in the field of project management and software development. This practice works on the continuous iteration of testing and development for the complete Agile software development lifecycle of a given business project.

Rather than making big changes at work, Agile aims to help teams work toward small achievable goals, which proves to be successful in delivering customers what they need easily without having to set big, unattainable goals. Agile requires professionals to continuously evaluate several plans, requirements, and results, and this allows them to respond to necessary changes easily and quickly.

**If you want to know more about Agile, you must check out the YouTube video below:**

**Core Values of Agile**

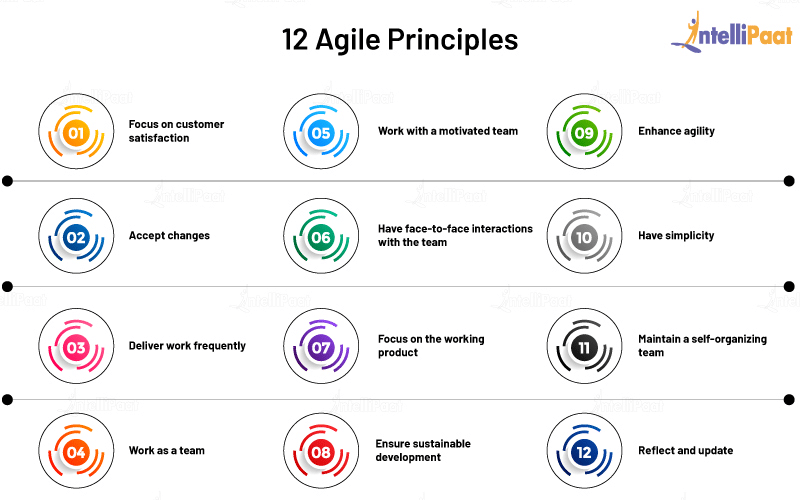
Before reading about the principles followed by Agile, let’s check out some core values of Agile that are mentioned in the Agile Manifesto:

* Individuals and interactions over tools and processes
* Working product over comprehensive documents
* Customer collaboration over contract negotiation
* Responding to changes over following a plan

Now, let us take a look at the 12 principles that are followed in Agile technology.

**Agile Principles**

* **Focus on customer satisfaction:**The first priority of Agile is to ensure customer satisfaction via quick and continuous delivery of necessary products.
* **Accept changes:** It is important to address the changes in requirements, even if they are introduced late in the development process. Agile implements changes to help companies gain a competitive advantage over their rivals.
* Deliver work frequently: The product needs to be delivered frequently, keeping in mind the shortest timescale for customer preference.
* Work as a team: The development team and the business team must work together for the entire duration of a given project.
* Work with a motivated team: While working with a team, ensure that the team members are trusted, motivated, and have faith in the project. You must also create an environment where the team members can work with each other and receive the support required.
* Have face-to-face interactions with the team: Having face-to-face conversations and interactions with team members is among the most effective methods for communicating and conveying the required information.
* Focus on the working product: The significant measure that shows progress is the working software; it indicates the progress toward the final result.
* Ensure sustainable development: Agile methods can promote sustainability when developers, sponsors, and users are able to maintain a regular and constant pace.
* Enhance agility: The agility of the product can be enhanced by providing regular attention to the technical details and design.
* Have simplicity: It is important to have simplicity in your product to waste minimal time, use most of the time efficiently, and get the work done.
* Maintain a self-organizing team: The best type of design and architecture can only be achieved through a team consisting of self-organizing members.
* **Reflect and update:**The team reflects on making the product more effective regularly. The team further adjusts and makes improvements in the features of the product.



**Agile Software Development**

Agile software development is a compiled set of practices and frameworks based on the core principles and values mentioned in the Agile Manifesto. Agile methodology is one of the best ways to approach software development compared to other available methods. One feature that makes the Agile methodology unique is that it requires team members to work together. Moreover, results are achieved through the collaboration of cross-functional, self-organizing teams.

Just because the teams are self-organizing does not mean that they do not require managers; they do. Managers ensure that the team members have the necessary skills required by the project; managers also offer and facilitate the required environment for the project. Generally, managers allow team members to identify how to deliver the product without interfering, but whenever team members have trouble coming up with solutions that can resolve the issue at hand, managers step in.

Now that you have read enough about Agile, its principles and core values, and its role in the field of software development, let us take a look at the various types of Agile methodologies that help in software development.

**What is Agile Methodology?**

In simple terms, Agile methodology in the field of project management is the process of managing a given business project, which is defined by using collaboration and iterations at regular intervals so that it meets the requirements of a particular customer. Agile uses various methods or frameworks, such as Scrum, [Kanban](https://intellipaat.com/blog/what-is-kanban/), and Extreme Programming, that can be implemented in the project to make it successful.

You will read about these and a few other Agile frameworks in the next section of the blog.

**Types of Agile Methodology**

There are numerous Agile methodologies that are used by most companies. A few of them are explained briefly below:

**Scrum**

Scrum is a framework used by team members to form a hypothesis, put it to the test, look back and reflect on it, and make necessary changes. This framework also allows professionals to adopt practices from other frameworks as per requirements. It makes the project move ahead through short-term work blocks called sprints. These sprints are often limited to an interval of two weeks.

**Kanban**

The Kanban framework helps in designing, managing, and making improvements in the flow of the system. Kanban allows companies to get a visual of their workflow and the work that is pending and needs to be done.

**Extreme Programming**

Extreme Programming allows team members to develop high-quality software that makes improvements in the overall work quality. It helps in developing software using specific engineering practices. Extreme Programming is of significant use when teams are dealing with the risk that comes with handling new changes that might have been caused because of various reasons.

**Lean**

Lean consists of a varied range of principles and tools that identify and eliminate waste to help speed up the development process. The Lean software development framework helps in maximizing value and minimizing waste. Most industries that produce waste during [SDLC](https://intellipaat.com/blog/what-is-sdlc/) use Lean to reduce it.

**Crystal**

Crystal is an Agile framework that manages interactions with customers instead of focusing on processes and tools. Crystal works on the dynamic and unique nature of the project and is responsible for streamlining tasks to make optimized improvements. It helps improve a team’s strength concerning continuous integration, communication, and so on.

After getting a detailed introduction to Agile, now you will read about the Waterfall model.

**Waterfall Model**

Like the Agile model, the **waterfall model** is also a method for software development. It is an old approach, in the field of SDLC, that requires software development teams to follow a laid-out process and move on to the next step only once the previous and current steps are completed. In the waterfall model, before the start of any new phase, the old phase must be successfully completed.

**Steps of the Waterfall Model**

The steps followed by professionals while using the waterfall model are as follows:

* **Requirement:** In this step, all the requirements of the system are gathered and documented for analysis.
* **System design:** The requirements gathered in the previous step are studied to set up the design of the system. This allows professionals to understand the hardware specifications and system requirements, along with defining the architecture of the system.
* **Implementation:** Here, small programs are built on the basis of system design. These programs are units that can be integrated into the upcoming stage in the model. The functionality of the units is then tested; this process is known as unit testing.
* **Testing:**Once a unit is tested, it is adopted by the system. In this testing and integration stage, after all, units are integrated, the entire system is checked for failures and faults.
* **Deployment:** After the completion of the testing phase, the product is released to the market and made accessible to the users.
* **Maintenance:** The last phase of the model is to deal with the issues that come up from clients after the product patches are released. Maintaining the project will help in making improvements and enhancing it. It will also allow you to deliver the changes to the client.

Further in this blog, you will read about some of the differences between the two software development methods.

**Agile Model vs Waterfall Model**

|  |  |
| --- | --- |
| **Agile Model** | **Waterfall Model** |
| The life cycle of project development is divided into sprints | The development process is divided into multiple phases |
| It follows an incremental method | It follows a sequential method |
| One of the benefits of Agile is its flexibility | It is a structured method of developing software, which often makes it rigid |
| It can be referred to as a collection of several projects | It is completed as a single project |
| The changes in the requirements of project development can be made even after the completion of the initial plan | There is no room for making adjustments to the requirements once you begin the development |

Let us read about the advantages of the Agile model and the disadvantages of the waterfall model that have led to the creation and adoption of Agile in many companies.

**Benefits of the Agile Model**

* It allows interaction and collaboration between the project team and the clients.
* It helps in providing transparency to the clients and offering them clear knowledge of the steps of project development.
* The product is always delivered on time or early.
* The product cost can be estimated.
* Changes help in making an improved product.
* The entire project is broken down into small parts that offer a better quality of development and testing.

**Drawbacks of the Waterfall Model**

* The actual software is created toward the end of the project life cycle.
* It involves a lot of uncertainty and risk as the testing is done once the project is completed.
* It is not suitable for long and complex projects that are object-oriented.
* It is difficult to measure the development progress in the various phases.
* It does not accommodate changes in the requirements.