**What is Agile Software Development Framework?**

An Agile Software Development framework refers to a collection of practices and principles that guide the development, execution, and management of software development projects. Agile Software Development Methodology enhance collaboration, customer feedback, and quick iterations.

**What is Agile?**

Agile methodology is a project management and software development approach that aims to be more effective.

1. It focuses on delivering smaller pieces of work regularly instead of one big launch.
2. This allows teams to adapt to changes quickly and provide value to customers faster.

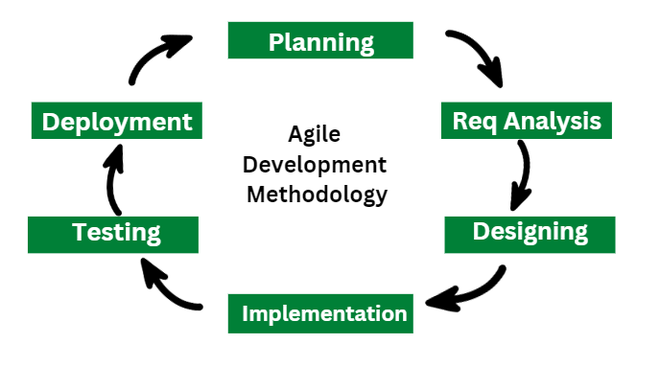
**What is the Agile Methodology?**

Agile methodologies are iterative and incremental, which means it’s known for breaking a project into smaller parts and adjusting to changing requirements.

1. They prioritize flexibility, collaboration, and customer satisfaction.
2. Major companies like Facebook, Google, and Amazon use Agile because of its adaptability and customer-focused approach.

## Life cycle of Agile Methodology

The Agile software development life cycle helps you break down each project you take on into six simple stages:



### 1. Requirement Gathering

* In this stage, the project team identifies and documents the needs and expectations of various stakeholders, including clients, users, and subject matter experts.
* It involves defining the project’s scope, objectives, and requirements.
* Establishing a budget and schedule.
* Creating a project plan and allocating resources.

### 2. Design

* Developing a high-level system architecture.
* Creating detailed specifications, which include data structures, algorithms, and interfaces.
* Planning for the software’s user interface.

### 3. Development (Coding)

Writing the actual code for the software. Conducting unit testing to verify the functionality of individual components.

### 4. Testing

This phase involves several types of testing:

1. **Integration Testing:** Ensuring that different components work together.
2. **System Testing:** Testing the entire system as a whole.
3. **User Acceptance Testing:** Confirming that the software meets user requirements.
4. **Performance Testing:**Assessing the system’s speed, scalability, and stability.

### 5. Deployment

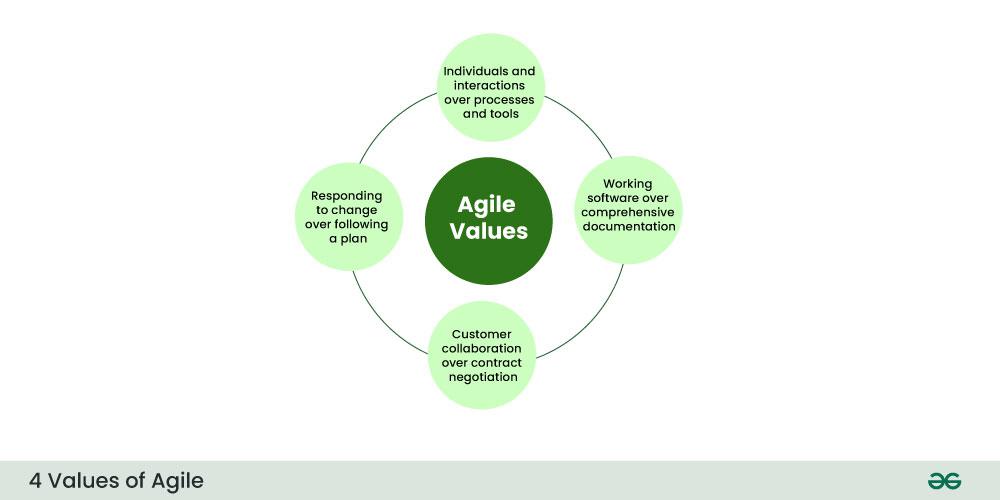
1. Deploying the software to a production environment.
2. Put the software into the real world where people can use it.
3. Make sure it works smoothly in the real world.
4. Providing training and support for end-users.

### 6. Review (Maintenance)

1. Addressing and resolving any issues that may arise after deployment.
2. Releasing updates and patches to enhance the software and address problems.

**4 Core Values of Agile Software Development**

The Agile Software Development Methodology Manifesto describe four core values of Agile in software development.

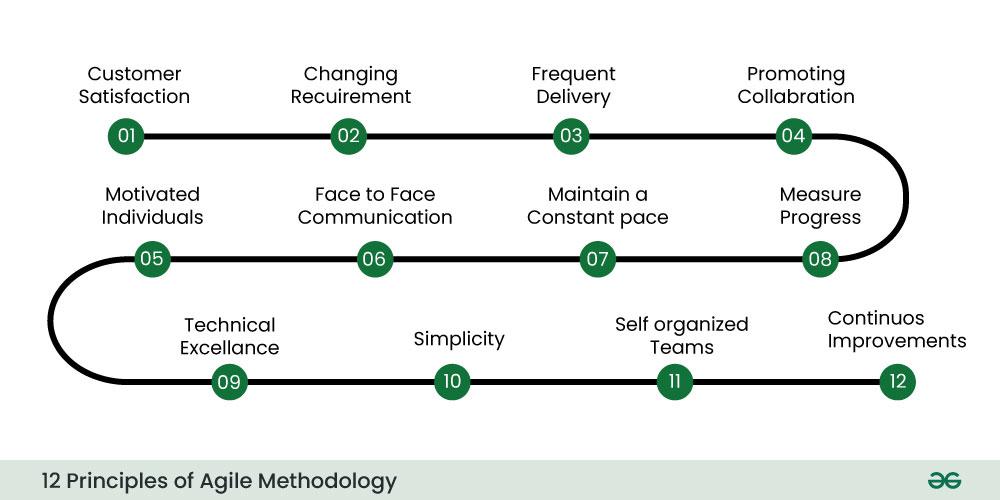


*4 Core Values of Agile Software Development Methodology*

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

**12 Principles of Agile Software Development Methodology**

The Agile Manifesto is based on four values and twelve principles that form the basis, for methodologies.



*12 Principles of Agile Software Development Methodology*

These principles include:

1. Ensuring customer satisfaction through the early delivery of software.
2. Being open to changing requirements in the stages of the development.
3. Frequently delivering working software with a focus on preference for timeframes.
4. Promoting collaboration between business stakeholders and developers as an element.
5. Structuring the projects around individuals. Providing them with the necessary environment and support.
6. Prioritizing face to face communication whenever needed.
7. Considering working software as the measure of the progress.
8. Fostering development by allowing teams to maintain a pace indefinitely.
9. Placing attention on excellence and good design practices.
10. Recognizing the simplicity as crucial factor aiming to maximize productivity by minimizing the work.
11. Encouraging self-organizing teams as the approach to design and build systems.
12. Regularly reflecting on how to enhance effectiveness and to make adjustments accordingly.

## When to use the Agile Methodology?

It is particularly well-suited for projects and organizations where the following conditions or needs are present:

1. **Unclear or Changing Requirements:** Agile is great for projects with requirements that aren’t well-defined or might change.
2. **Complex Projects:** It’s good for big, complex projects by breaking them into smaller pieces.
3. **Customer Focus:** Use Agile when making customers happy is a priority and you want to involve them throughout.
4. **Quick Time-to-Market:** If you need to get your product out fast, Agile can help.
5. **Small to Medium Teams:**Agile works well for teams of a few to a few dozen people.
6. **Team Skills:** It’s best when you have a mix of skills in your team, like development, testing, design, and more.
7. **Collaboration:** Agile promotes working together and open communication.
8. **Regular Updates:** If you want to check progress often and make changes as needed.
9. **Transparency:** Agile emphasizes being open and clear with everyone involved in the project.
10. **Risk Control:** It helps manage risks by tackling issues as they come up.
11. **Innovation:** If you encourage trying new things and learning from experience, Agile supports that.
12. **Continuous Improvement:**Agile fosters a culture of always getting better over time.

**Popular Agile Software Development Frameworks**

There are Agile frameworks that have gained recognition in the software development world. Some of the known ones include:

**Scrum:**

Scrum is the type of Agile Framework, widely known for its incremental approach. It involves sprint cycles, daily stand-up meetings, and a focus, on product backlogs.

**Features of Scrum in software development:**

* Scrum is light-weighted framework.
* Scrum emphasizes self-organization.
* Scrum is simple to understand.
* Scrum frameworks help the team to work together.

