Agile Software Development Methodology | Framework, Principles, and Benefits

Agile Software Development Methodology in software development is an efficient methodology that helps teams produce high-quality software quickly and with flexibility. Agile is not just a methodology; it's a mindset. At its core, Agile values individuals and interactions, working solutions, and customer collaboration over strict processes and comprehensive documentation. It acknowledges that the needs and priorities of a project may change, emphasizing the importance of adaptability and continuous improvement.

What is Agile Software Development Framework?

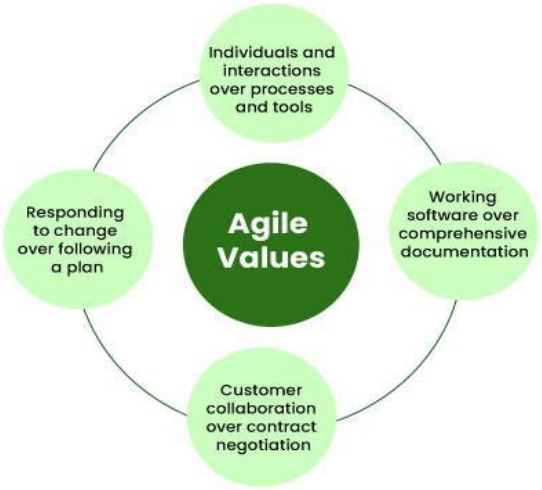
An Agile Software Development framework refers to a collection of practices and principles that guide the development, execution, and management of software development projects. Agile Software Development Methodology enhance collaboration, customer feedback, and quick iterations. Unlike other software development methods, Agile does not offer a solution; instead, it offers various frameworks customized to specific industries, project types, and team sizes.

What is Agile Methodology in Software Development?

Agile Software Development Methodology is a project management approach that allows successful and efficient execution of the project while emphasizing the improvement of a project and team collaboration. The approach is applicable in software development for flexibility, customer satisfaction, and collaboration. It refers to the application of a set of principles that functions through an interactive and incremental approach. The Agile Software Development Methodology emphasizes the importance of team collaboration and delivering a working product quickly to meet customer needs and expectations.

4 Core Values of Agile Software Development

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



4 Values of Agile

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
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 main 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.

Benefits of Agile Software Development
The Agile framework provides benefits, such, as;

Increased flexibility: Agile allows for adapting requirements and priorities to ensure that the final product meets customer needs.
Time to market: By using iterations and incremental development valuable features or products can be delivered faster.
Enhanced quality: Continuous feedback and testing together contribute to the production of higher quality deliverables.
Improved collaboration: Agile promotes collaboration and communication among functional teams fostering better teamwork.
Heightened customer satisfaction: Regular feedback, from customers drives product improvements and ensures customer satisfaction.

Agile Development Models – Software Engineering
In earlier days, the Iterative Waterfall Model was very popular for completing a project. But nowadays, developers face various problems while using it to develop software. The main difficulties included handling customer change requests during project development and the high cost and time required to incorporate these changes. To overcome these drawbacks of the Waterfall Model, in the mid-1990s the Agile Software Development model was proposed.

Agile Software Development
Agile Software Development is the software development process used to design complicated software. It is used when the software is quite sensitive and complicated. It is used when security is much more important. It is used by professionals to develop the software.

Advantages of Agile Software Development

Flexibility: Agile software development is highly flexible and can easily adapt to changes in requirements, design, and scope.
Customer Involvement: Agile software development encourages frequent customer involvement, which can result in a final product that better meets their needs.

Continuous Delivery: Agile software development typically includes continuous delivery, which means that working software is delivered to the customer on a regular basis.
Collaboration: Agile software development emphasizes collaboration between team members, which can lead to better communication and problem-solving.
Early and Frequent Testing: Agile software development includes early and frequent testing, which can help to catch issues and bugs early in the development process.

Disadvantages of Agile Software Development

Lack of Predictability: Agile software development can be less predictable than traditional methods, with less certainty about the final product and its delivery schedule.
Limited Documentation: Agile software development often relies less on documentation, which can make it difficult to track changes and understand the system architecture.
Time and Resource Constraints: Agile software development requires a significant commitment of time and resources from all team members.
Less Emphasis on Planning: Agile software development often places less emphasis on detailed planning, which can result in scope creep and delays.
Resistance to Change: Agile software development requires a significant cultural shift and may be difficult for some team members and organizations to adopt.

Difference Between Traditional and Agile Software Development

Traditional Software Development	Agile Software Development
It is used to develop simple software.	It is used to develop complicated software.
In this methodology, testing is done once the development phase is completed.	In this methodology, testing and development processes are performed concurrently.
It follows a linear organizational expectation structure.	It follows an iterative organizational structure.

Traditional Software Development	Agile Software Development
It provides less security.	It provides high security.
Client involvement is less as compared to Agile development.	Client involvement is high as compared to traditional software development.
It provides less functionality in the software.	It provides all the functionality needed by the users.
It supports a fixed development model.	It supports a changeable development model.
It is used by freshers.	It is used by professionals.
Development cost is less using this methodology.	Development cost is high using this methodology.
It majorly consists of five phases.	It consists of only three phases.
It is less used by software development firms.	It is normally used by software development firms.
The expectation is favored in the traditional model.	Adaptability is favored in the agile methodology.
Traditional software development approaches are formal in terms of communication with customers.	Agile software development methodologies are casual. In other words, customers who work with companies that utilize Agile software development approaches are more likely to interact with them than customers who work with companies that use traditional software

Traditional Software Development	Agile Software Development
	development methodology.
For starters, typical software development approaches employ a predictive approach. There is full specification and prediction of the software development processes because the product is produced through rigorous and explicit planning. Changes are not permitted in this technique because the time and cost of project development are fixed.	Here, a flexible approach is used as the software development approaches are founded on the notion of continual design improvement and testing relies on team and client feedback.
Examples Office productivity suites Data management software Media players Security programs	Examples Sky Phillips JP Morgan Chase
Models based on Traditional Software Development- Spiral Model Waterfall Model V Model Incremental Model	Models based on Agile Software Development- Scrum Extreme Programming (XP) Crystal Dynamic Systems Development Method (DSDM) Feature Driven Development (FDD) Adaptive Software Development (ASD)