Agile Methodology

Introduction about Agile

Agile methodology is an iterative and incremental approach to software development that emphasizes flexibility, collaboration, and continuous improvement. It focuses on delivering working software quickly and adapting to changing requirements and feedback from stakeholders. The Agile methodology involves breaking down a project into smaller, more manageable parts called sprints or iterations, which typically last for two to four weeks.

Different frameworks in Agile

There are several frameworks in Agile methodology that software development teams can use to plan, execute, and deliver projects. Here are some of the most commonly used Agile frameworks:

a. Scrum: Scrum is a framework that emphasizes teamwork, collaboration, and iterative progress towards a well-defined goal. It includes a set of ceremonies such as sprint planning, daily stand-up meetings, sprint reviews, and retrospectives.

b. Kanban: Kanban is a framework that focuses on visualizing work, limiting work in progress, and improving the flow of work. It emphasizes continuous delivery and encourages teams to work in small, incremental steps.

c. Extreme Programming (XP): XP is a framework that emphasizes coding and testing, customer involvement, and frequent releases. It includes practices such as pair programming, test-driven development, and continuous integration.

d. Lean Software Development: Lean software development is a framework that emphasizes reducing waste, optimizing flow, and maximizing customer value. It includes practices such as value

stream mapping, pull-based production, and continuous improvement.

e. Agile Unified Process (AUP): AUP is a framework that combines the principles of Agile methodology with the best practices of software development. It includes practices such as iterative development, risk management, and architecture-centric approach.

f. Crystal: Crystal is a framework that emphasizes people and communication over processes and tools. It includes practices such as incremental delivery, active user involvement, and automated tests.

g. Dynamic Systems Development Method (DSDM): DSDM is a framework that focuses on delivering projects on time and within budget while meeting the business needs. It includes practices such as timeboxing, iterative development, a continuous user involvement.

h. Feature Driven Development (FDD): FDD is a framework that emphasizes the timely delivery of small, client-valued features. It includes practices such as domain object modelling, feature lists, and regular progress reporting.

i. Scaled Agile Framework (SAFe): SAFe is a framework that supports the implementation of Agile methodology in large enterprises. It includes practices such as program increment planning, solution and portfolio management, and lean budgeting.

j. Disciplined Agile Delivery (DAD): DAD is a framework that combines the principles of Agile methodology with the disciplines of project management. It includes practices such as continuous improvement, disciplined agile governance, and explicit tailoring.

k. Nexus: Nexus is a framework that supports the implementation of Scrum in large, complex projects. It includes practices such as

scaling Scrum, managing dependencies, and coordinating multiple Scrum teams.

l. Large-Scale Scrum (LeSS): LeSS is a framework that supports the implementation of Scrum in large, multi-team environments. It includes practices such as coordinated sprints, shared product backlog, and area product owners.

m. Agile Project Management (Agile): Agile is a framework that combines the principles of Agile methodology with project management practices. It includes practices such as iterative development, timeboxing, and prioritized requirements.

n. Agile Coaching Framework (ACF): ACF is a framework that focuses on developing the skills and competencies of Agile coaches. It includes practices such as coaching skills, mentoring, and facilitating Agile adoption.

o. Agile Modelling (AM): AM is a framework that focuses on modelling and documentation in an Agile environment. It includes practices such as lightweight modelling, continuous feedback, and test-driven development.

Each Agile framework has its own strengths, weaknesses, and suitability for different types of projects and organizations. Choosing the right framework requires careful consideration of the project's context, goals, and constraints, as well as the team's skills and experience.

Agile Principles

The Agile methodology is based on a set of 12 principles that guide the development of software in an iterative and incremental manner. These principles are outlined in the Agile Manifesto, which was developed by a group of software developers in 2001. Here are the 12 principles of Agile:

1. Customer satisfaction 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, with a preference to the shorter timescale.

4. Businesspeople 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.

These Agile principles promote a collaborative, iterative, and customer-centric approach to software development. By prioritizing

customer satisfaction, change management, frequent delivery, team collaboration, and continuous improvement, Agile methodology enables teams to deliver high-quality software quickly and efficiently.

Agile Values

The Agile methodology is based on a set of values that guide the development of software in an iterative and incremental manner. The Agile values are described in the Agile Manifesto, which was developed by a group of software developers in 2001. Here are the four Agile values outlined in the Agile Manifesto:

1. Individuals and interactions over processes and tools: This principle emphasizes the importance of people and their communication in software development. Agile methodology values face-to-face communication, collaboration, and teamwork over rigid processes and tools.

2. Working software over comprehensive documentation: This principle emphasizes the importance of delivering working software that meets the customer's needs over extensive documentation. Agile methodology values working software as the primary measure of progress and encourages teams to keep documentation to a minimum.

3. Customer collaboration over contract negotiation: This principle emphasizes the importance of working closely with customers to understand their needs and to deliver software that meets those needs. Agile methodology values customer collaboration and feedback throughout the development process over detailed contracts and negotiations.

4. Responding to change over following a plan: This principle emphasizes the importance of adapting to change and responding to new requirements as they arise. Agile methodology values

flexibility and the ability to change direction in response to feedback over rigid adherence to a plan.

These Agile principles promote a collaborative, iterative, and customer-centric approach to software development. By prioritizing individuals, working software, customer collaboration, and responsiveness to change, Agile methodology enables teams to deliver high-quality software quickly and efficiently.

Scrum Framework

Scrum is an Agile framework for managing and completing complex projects. It is a lightweight, iterative approach that enables teams to deliver valuable software incrementally and adapt to changing requirements. The Scrum framework consists of three main roles, five events, and three artifacts.

What is scrum?

Scrum meaning - Scrum is a framework for developing, delivering, and sustaining complex products. It is essentially a framework wherein people can address complex, adaptive problems. At the same time, they can deliver products of the highest possible value in a productive and creative manner.

Define Scrum - What does Scrum mean?

Scrum is an iterative and incremental structure for project management mainly used in agile software development. The scrum methodology indicates functional software, the versatility to change accompanying with emerging communication, collaboration, and business realities.


Daily Stand Up
24 H
Product Owner
Dev Team
Scrum Master
Sprint 1–4 Weeks
Sprint Review + Sprint Retrospective
Product Backlog
Sprint Planning Meeting
Sprint Backlog
Finished Work