

Share on your Social
Media



Agile Tutorial for Freshers and Professionals

Published On: July 29, 2024

Agile Tutorial for Freshers and Professionals

When it comes to software development, the Agile methodology places a strong emphasis on teamwork and meeting customer expectations by delivering a functional product on time. This Agile tutorial helps you understand the complete concepts of Agile, along with how it works in software development.

[Download Agile Tutorial PDF](#)

Introduction to Agile

The Agile technique is a project management style that prioritizes ongoing collaboration and improvement while segmenting the work into manageable chunks.

We cover the following in our Agile tutorial:

- Agile Development Concepts
- Agile Manifesto
- Characteristics of Agile
- Daily Stand-Up
- Definition of Done
- Release Planning

Featured Articles



Want to know
more about
becoming an
expert in IT?

[Click Here to Get Started](#)

100%
Placement
Assurance

AUTHORISED
CERTIFICATION
PARTNER

IBI

Related Courses at SLA

Related Posts



C and C++ Tutorial

Published On: August 1, 2024

C and C++ Tutorial C is a high-level, procedural, general-purpose programming language. Whereas C++, a...



Quick Enquiry

- Iteration Planning
- Product Backlog
- Popular Terminologies of Agile

Agile Interview Questions

Important Agile Development Concepts

The following are the main concepts of agile software development:

Adaptability: The project must be flexible to meet the ever-evolving demands of the development process.

- It saves time, boosts team confidence, helps the project get back on track, and reduces productivity because of the mess.
- To provide results on schedule, it also permits modifying the development and deployment paths.

Disruption: The agile technique promotes adaptability in order to increase customer satisfaction and produce high-quality results.

- It involves adjusting goals, priorities, and culture.
- Even though the changes are difficult and unpleasant, managing collective agreements improves performance while becoming simpler.

Collaboration: It is yet another highly regarded Agile idea that supports the development of responsibility and trust. It facilitates the exchange of ideas and group debate to enhance weak but potentially viable concepts.

Iterative Developments: In software development, rework for enhancements is a typical occurrence. Here, the iterative method adheres to a close-knit feedback loop for ongoing progress. It is the fundamental idea of agile development.



ASP DOTNET Tutorial

Published On: July 31, 2024

ASP DOTNET Tutorial Microsoft created the web framework known as ASP.NET. It is employed in...



Artificial Intelligence Tutorial

Published On: July 30, 2024

Artificial Intelligence Tutorial Artificial intelligence (AI) is significant since it enhances many facets of society...



Appium Testing Tutorial

Published On: July 30, 2024

Appium Testing Tutorial Designed to make the UI automation of many app platforms easier, Appium...

Incremental Development: It entails developing a rudimentary software version that has all the necessary functionalities.

- As voted upon during the conversations or meetings, more enhancements are done.
- The components of the software are updated, and new features that are easy to use are added.

Milestone Checks: Its goal is the growth and development of both clients and teams. Once the teams reach a particular and important point in the project, they examine the earlier advancements.

Agile Training

Agile Manifesto

A group of 17 software engineers gathered to talk about lightweight software development techniques. Following their meeting, the following resulted.

Twelve Principles of Agile Manifesto

An Agile Software Development Manifesto contains 12 principles:

Customer satisfaction: Delivering valuable software on time and consistently is our top goal in order to meet customer requirements.

Acceptable Change: At this late stage of development, requirements that are constantly evolving should be welcomed. Agile procedures ought to improve clients' competitive edge.

Provide Functional Software: Provide functional software on a regular basis, within a few weeks to a few months, taking into account a shorter timeline.

Collaboration: Throughout the course of a project, business professionals and developers must cooperate.

Motivation: Establish a trusting and supportive environment for each team member to help them feel accountable for completing the task at hand.

Face-to-Face Communication: In a development team, face-to-face communication is the most effective and efficient way to communicate information.

Assess Progress Based on Working Software: The most important and appropriate way to gauge progress is through working software.

Keep Up a Constant Pace: Agile methods strive for long-term growth. The project's business, developers, and consumers should all be able to keep up a steady pace.

Monitoring: To improve agility, consistently focus on excellent technical design and strong engineering.

Simplicity: To measure the labor that is not finished, keep things simple and use simple language.

Self-organized Teams: Since self-organized teams provide the best architectures, requirements, and designs, agile teams ought to be self-organized and not rely too much on outside teams.

Examine the Work Often: Examine the completed work on a frequent basis to help the team think through ways to improve and modify its approach as necessary.

[Agile Course Syllabus PDF](#)

Characteristics of Agile Methodology

Several significant features of the product produced using the agile technique are listed below.

Fixed-Length Iterations

The two main delivery units of the agile software development methodology are release and

iteration.

- A single version is made up of multiple phases of iterations. Every iteration is made up of a single microproject.
- Various agile development tasks, such as organizing, estimating, and prioritizing defects, enhancement requests, and other work items, are assigned to a release.

Working, Tested Software

Delivering functional, tested feature software is the agile development team's main goal. Enabling and enhancing client feedback is based on functional characteristics.

- It facilitates overall project visibility as well as team collaboration.
- They offer this proof to ensure that the project and the system are moving forward.
- The most recent feedback from users, clients, and other stakeholders is used in this process.
- The team consistently collaborates to identify the optimal business solution at every stage of the product development process.

Value-Driven Development

The early and ongoing delivery of business value is the primary goal of the agile development process. Software that has been tested is used to measure it.

- Product features are the primary focus of the development team and serve as the main organizing, monitoring, and delivery unit.
- The team keeps tabs on the number of products in use and the tested features they are delivering as they progress from one iteration to the next.

Continuous Planning

Continuous planning relies on a single iteration.

- As the project gets underway, the development team only makes further plans to kick off the first iteration and, if necessary, to outline a high-level feature delivery schedule.
- Before the iteration begins, the team decides which features to implement and calculates the technical tasks associated with each feature.

Multi-level Planning

If continuous planning takes place on at least two levels, the results are even more significant:

- The development team decides which features are essential, desirable, and feasible to complete by the deadline at the release level.
- The development team selects and prioritizes the next set of features to be implemented at the iteration level.

The development team breaks down product features further if they are too large to predict or deliver in a single iteration.

Relative Estimation

The idea of relative estimating or/and predetermined estimation buckets saves discussion time and helps the team work together more efficiently. When a product feature goes over a mutually agreed-upon maximum estimate, it ought to be divided into more features.

- Many agile development teams utilize relative feature estimation as a way to speed up planning. It gets rid of superfluous complexity.
- The development team estimates all features using a subset of three to five relative estimation categories, or buckets.

Emergent Feature Discovery

Agile development projects prioritize and estimate features rapidly, refining the specifics as needed, as opposed to taking weeks or months to analyze the

requirements before beginning development.

When customers, testers, and developers collaborate, the product feature is explained in greater detail.

Continuous Testing

By employing continuous testing, we can lower the project's risk of failure. We assess the software product's progress and avert errors through continuous testing. We take care of the functional and verified aspects.

Continuous Improvement

It is associated with continuous improvement and testing happens together.

If any problems or project failures are discovered during continuous testing, we continuously fix those issues right away. We are always improving the system and the project.

Small, Cross-functional Teams

With each iteration, an incremental software product is delivered. For the development teams to successfully produce valuable software, cross-functionality is another requirement.

Daily Stand-Up

An agile team's daily stand-up is a status meeting with every member.

What is the Daily Stand-Up?

Every team member gathers for a daily status meeting known as a "stand-up," which lasts for around fifteen minutes.

- Each participant must respond to three crucial inquiries:
 - What did I carry out yesterday?
 - What am I going to do today?
 - Anything preventing me from or

preventing me from accessing.

- Daily stand-ups are for status updates only; discussions are not allowed. Team members ought to arrange a second meeting at a different time for discussion.
- Typically, attendees stand rather than sit in order to expedite the conclusion of the meeting.

Why is daily stand-up important?

The following are the advantages of an agile daily stand-up:

- Every day, the team can assess their work and determine whether they are on track to meet the iteration plan's objectives.
- Every team member updates the group on their daily duties.
- It gives the team visibility into any delays or roadblocks.

Who shows up for a stand-up?

Every day, stand-ups should be attended by the delivery team, the scrum master, and the product owner.

- Although they are welcome to witness the meeting, stakeholders and customers are not permitted to take part in stand-ups.
- The scrum master is in charge of recording the questions and issues that each team member has.

Tips to implement daily stand-up

There are various methods to conduct stand-ups when members of an agile team are located in different time zones.

- Choose a team member who can join the stand-up meeting for teams that are located in different time zones on a rotating basis.
- Each team should have its own stand-up, and

the status of each should be updated in a platform like Rally, SharePoint, Wikis, etc.

- Prepare a wide range of communication tools, such as instant messaging, conference calls, video conferencing, and other third-party knowledge sharing platforms.

Definition of Done in Agile

The three phases of the agile definition of done are product release, iteration, and user story (requirement). These are listed below:

User Story (requirement)

A need that is condensed into a few phrases is called a user narrative. The user's common language is a necessity.

The goal of this iteration is to finish this user story. When the user story is finished,

- The relevant documentation and code have all been checked in.
- The product successfully completed each step of the unit test.
- The acceptance test case's entire set of procedures has been relocated.
- The product owner needed to provide their approval to the story.
- The documentation, or help text, has been written.

Iteration

A time-based collection of user stories is called an iteration. It fixes the flawed product and is approved for release along with the product.

The definition of an iteration is made during the planning meeting and finalized during the demo and review meeting. Another name for the iteration is a sprint.

Repetition is necessary in the following situations:

- The product's performance has been evaluated.
- The product backup has finished.
- The user's request has been moved or approved for the upcoming iteration.
- The defective product has been corrected or is being held off until a later version.

Release

The product release, a momentous occasion, represents the delivery of work both inside and outside. The system or product version, is also tested. The release of the product occurs when:

- Stress tests are performed on the system.
- High performance is achieved.
- incorporate the product's security validation.
- A disaster recovery plan is put to the test.

Release Planning

The plan for delivering a product increment is what release planning is all about. Every two to three months, it is completed.

Who takes part in this?

Scrum Master: The scrum master supports the agile delivery team by serving as a facilitator.

Product Owner: The product owner embodies the overall perspective of the backlog of products.

Agile Team: The agile delivery team offers valuable information regarding technological feasibility and dependencies.

Stakeholders: When decisions are being made regarding the release planning, stakeholders such as customers, program managers, and subject matter experts serve as advisors.

Iteration Planning

The team's goal in iteration planning is to finish the

collection of highest-priority product backlog items. The team velocity and iteration length set a limit on the commitment's duration.

Who takes part in this?

Scrum Master: The scrum master supports the agile delivery team by serving as a facilitator.

Product Owner: The product owner handles the acceptance criteria and a comprehensive picture of the product backlog.

Agile Team: Agile delivery outlines their responsibilities and determines the amount of work needed to complete the assignment.

Product Backlog

A backlog of unfinished products is called a product backlog. Features are described and ranked for each item. Ideally, things ought to be divided into user stories.

Why is the backlog of products important?

- It is ready so that each and every feature can have an estimate assigned to it.
- It facilitates the creation of the product's roadmap.
- Reordering the features makes it possible to add more value to the product.
- It is useful in figuring out what should come first. The team assigns a value to the item after ranking it.

Features of the Product Backlog

- Every product should have its own backlog, which may contain a number of features ranging from huge to extremely enormous.
- On a single product backlog, many teams can collaborate.
- Features are ranked according to their strategic suitability, business value, technical

value, or risk management.

- To be finished in later iterations, the highest ranked items are broken down into smaller stories during release preparation.

Popular Terminologies of Agile

Acceptance Standards: These are the parameters that the client or the product owner sets in order to accept a feature as legitimate and compliant with their needs.

Backlog Grooming: Agile teams provide feedback to the product manager or customer, who maintains the product backlog continuously.

The portfolio items must be prioritized, broken down into smaller components, planned for future iterations, new stories must be created, acceptance criteria must be updated, or acceptance criteria must be thoroughly explained.

Capacity: It is the quantity of work that a group can finish in a single iteration.

Feature: A product or capability that has been improved is valuable to stakeholders and can be released.

Iteration: Iteration planning defines the work to be done throughout each iteration, which concludes with a demo and review meeting. Another name for it is Sprint.

Increment: An increment is a product's evolving state as it goes through a phased development process. Typically, it is expressed in terms of set iterations or milestones.

Product Owner: As a member of the Agile delivery team, the product owner is in charge of gathering and prioritizing business needs for the product backlog.

Product Backlog: A collection of both functional and non-functional product needs.

Product Backlog Items: Examples of things the Agile team may need to build are features, problems, and user stories.

Points: A standard unit of measurement for determining the proportional size of features, user stories, and other portfolio components.

Release: A time period in which work is completed to enable software delivery of tested increments. A release in scrum is made up of several iterations.

Requirement: A software product specification meant to fulfill a specified function or contract. Requirements include things like user stories and portfolio items.

Story Points: A measurement that the agile team makes to determine the proportional sizes of features and user stories.

Timebox: A set amount of time allotted to the development of a deliverable. Typically, a timebox's start and end dates are fixed, and its resource count is as well.

Task: Each task can be assigned to a team member, who can then revise estimations, log completed work, or do as they like.

User Story: A set of acceptance standards to meet specific user needs. Usually, it is written from the viewpoint of the final user.

Velocity: A method for weighing approved work within a timebox or iteration. It is often the total of the story points that are approved in a given iteration.

Conclusion

Our Agile tutorial covers everything you need to know about the software development lifecycle that

implements Agile methodologies. Hone your skills
with our **[Agile training in Chennai.](#)**

Share on your Social
Media



Softlogic Academy

Softlogic Systems

KK Nagar [Corporate Office]

No.10, PT Rajan Salai, K.K. Nagar, Chennai
– 600 078.

Landmark: Karnataka Bank Building

Phone: [+91 86818 84318](tel:+918681884318)

Email: enquiry@softlogicsys.in

Map: [Google Maps Link](#)

OMR

No. E1-A10, RTS Food Street
92, Rajiv Gandhi Salai (OMR),
Navalur, Chennai – 600 130.

Landmark: Adj. to AGS Cinemas

Phone: [+91 89256 88858](tel:+918925688858)

Email: info@softlogicsys.in

Map: [Google Maps Link](#)

Courses

Python

Software Testing

Navigation

[About Us](#)

[Blog Posts](#)

[Careers](#)

[Contact](#)

[Placement Training](#)

[Corporate Training](#)

[Hire With Us](#)

[Job Seekers](#)

[SLA's Recently Placed Students](#)

[Reviews](#)

[Sitemap](#)

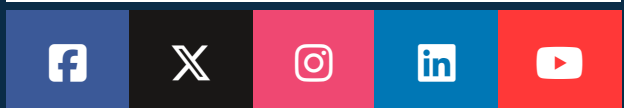
Important Links

[Disclaimer](#)

[Privacy Policy](#)

[Terms and Conditions](#)

Social Media Links



Full Stack Developer

Java

Power BI

Clinical SAS

Data Science

Embedded

Cloud Computing

Hardware and Networking

VBA Macros

Mobile App Development

DevOps

Review Sources

Google

Trustpilot

Glassdoor

Mouthshut

Sulekha

Justdial

Ambitionbox

Indeed

Software Suggest

Sitejabber

Copyright © 2024 – Softlogic
Systems. All Rights Reserved

SLA™ is a trademark of Softlogic Systems, Chennai.
Unauthorised use prohibited.