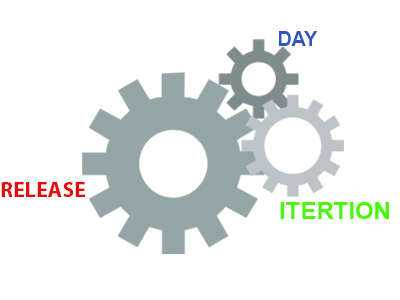
Agile Characteristics

The product developed under agile methodology has seen several important characteristics that are given below.

Agile Development Releases and Fixed-Length Iterations

The agile software development method is based on two central units of delivery: release and iteration. A single version consists of several iterations phase. Each iteration consists of its micro-project. The different functions of agile development like defects, enhancement requests and other work items are organized, estimated, and prioritized, and then assigned to release.



Agile Development Delivers-Working, Tested Software

The primary measure of the agile development team is to deliver working, progress and tested feature software. Working features serve as the basis for enabling and improving customer feedback. It also serve as team collaboration, and overall project visibility. They provide such evidence so that both the system and the project are on track.

At every step of product development, the team continuously works to assemble on the best business solution. This is done using the latest input from users, customers, and other stakeholders.

Value-Driven Development

Agile development methodology focuses really on delivering business value early and continuously. It is measured by running tested software. The development team focuses on product features as the central unit of planning, tracking, and delivery.

As the development goes on from iteration to iteration, the team tracks how many product are running, tested features they are delivering.

Continuous (Adaptive) Planning

As the project launches, the development team does just more planning to get going with the initial iteration and, if it is appropriate, to lay out a high-level release plan of features. The single iteration leads the key to continuous planning.

As the iteration starts, the team choose a set of features to implement, determines and estimates each technical task for each feature.

Multi-Level Planning in Agile Development

The continuous planning impacts much more significant result if it occurs on at least two levels:

* At the release level, the development team identifies and prioritize the features they must have, would like to have, and they can do within the deadline.
* At the iteration level, development team picks and plan for the next batch of features to implement, in priority order. If the product features are too large to estimated or delivered within a single iteration, the development team break them down further.

Relative Estimation

Several agile development teams use the practice of relative estimation for features to accelerate planning. It removes unnecessary complexity. The development team selects a few (3-5) relative estimation categories, or buckets, and estimates all features in terms of these categories.

The concept of relative estimation or/and predefined estimation buckets that prevent the team from wasting time on debating. When the product feature exceeds an agreed maximum estimate, then it should be further broken down into multiple features.

Emergent Feature Discovery

As disputed to spending weeks or months, analyzing the requirements before initiating development, agile development projects quickly prioritized and estimated features, and then refine the details when required. The feature of the product is described in more detail between customers, testers, and developers working together.

Continuous Testing

Using continuous testing of software product, we determine the progress and prevent defects. We handle the running and tested features. Using continuous testing, we can reduce the failure risk in the project.

Continuous Improvement

Continuous testing and constant improvement are correlated with each other. While continuous testing, if we found any bugs or project failure, we continuously improve that bugs immediately. We continuously refine both the project and the system.

Small, Cross-functional Teams

The incremental software product is delivered at every iteration. The development teams must also be cross-functional to be successful in developing the valuable software.