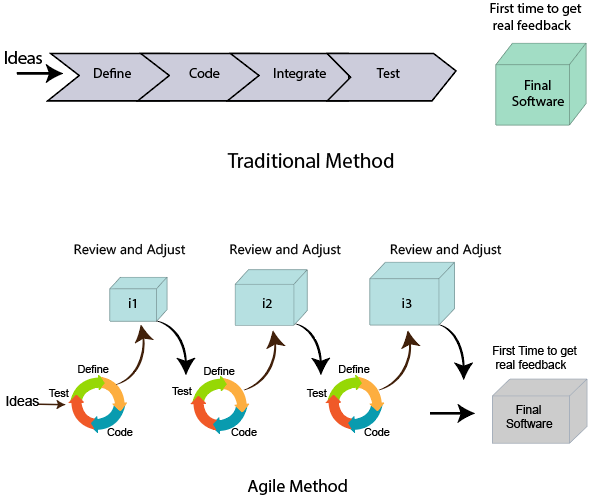
What is Agile Methodology?

An agile methodology is an iterative approach to software development. Each iteration of agile methodology takes a short time interval of 1 to 4 weeks. The agile development process is aligned to deliver the changing business requirement. It distributes the software with faster and fewer changes.

The single-phase software development takes 6 to 18 months. In single-phase development, all the requirement gathering and risks management factors are predicted initially.

The agile software development process frequently takes the feedback of workable product. The workable product is delivered within 1 to 4 weeks of iteration.



Roles in Agile

There are two different roles in a Agile methodology. These are the Scrum Master and Product Owner.

1. Scrum Master

The Scrum Master is a team leader and facility provider who helps the team member to follow agile practices, so that the team member meets their commitments and customers requirements. The scrum master plays the following responsibilities:

* They enable the close co-operation between all the roles and functions.
* They remove all the blocks which occur.
* They safeguard the team from any disturbances.
* They work with the organization to track the progress and processes of the company.
* They ensure that Agile Inspect & Adapt processes are leveraged correctly which includes
  + Planned meetings
  + Daily stand-ups
  + Demo
  + Review
  + Retrospective meetings, and
  + Facilitate team meetings and decision-making process.

2. Product Owner

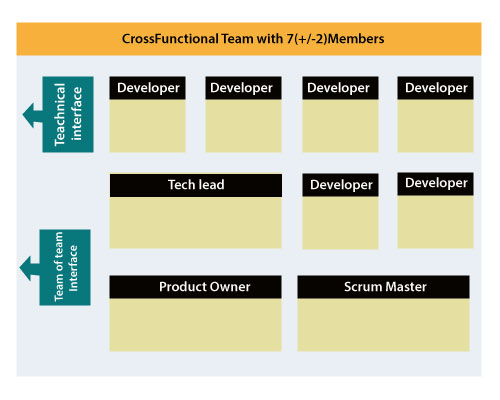
The Product Owner is one who runs the product from a business perspective. The Product Owner plays the following responsibilities:

* He defines the requirements and prioritizes their values.
* He sets the release date and contents.
* He takes an active role in iteration and releasing planning meetings.
* He ensures that the team is working on the most valued requirement.
* He represents the voice of the customer.
* He accepts the user stories that meet the definition of done and defined acceptance criteria.

Cross-functional team

Every agile team contains self-sufficient team with 5 to 9 team members. The average experience of each member ranges from 6 to 10 years. The agile team contains 3 to 4 developers, 1 tester, 1 technical lead, 1 scrum master and 1 product owner.

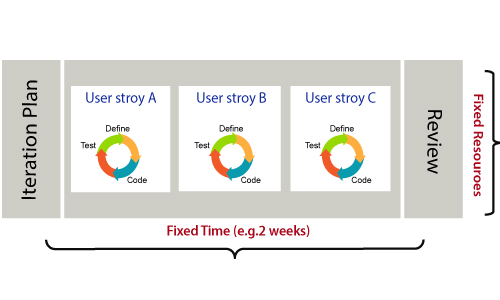
The Scrum master and Product owner are considered as a part of Team Interface, on the other hand remaining members are the part of Technical Interface.



How an Agile Team plan their work?

An Agile methodology is not a specific set of ceremonies or specific development techniques. Rather, it is a group of methodologies that demonstrate a commitment to tight feedback cycles and continuous improvement. An Agile team works in iterations to deliver the customer requirement, and each iteration takes 10 to 15 days. However, the original Agile Manifesto didn't set the time period of two-week iterations or an ideal team size.

Each user requirement is a planned based and their backlog prioritization and size. The team decides, how much scope they have and how many hours available with each team to perform their planed task.



What is a user requirement?

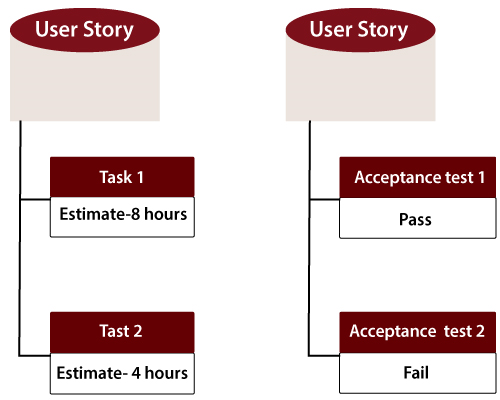
The user requirement defines the requirements of the user in terms of functionalities. There may be of two type of functionality.

* As a <User Role> I want <Functionality> so that <Business Value>
* In order to <Business value> as a <User Role> I want <Functionality>.

During software release planning, a rough estimate is given to a user requirement using relative scale points. During iteration planning, the requirement is broken down into tasks.

Relation between User requirement and Task

* User requirement talks about what is to be done. It defines the needs of users.
* Task talks about how it is to be done. It defines how functionality is implemented.
* User requirements are implemented by tasks. Every requirement is gathering as the task.
* User requirement is divided into different tasks when it is planned in current iteration.
* User tasks are estimated in hours based, generally it is between 2 to 12 hours.
* Requirements are validated using acceptance test.



When the requirement is completed

The Agile team decides the meaning of task done. There may be different criteria for it:

* When the entire task (development, testing) are completed.
* When all the acceptance tests are running and are passed.
* When no defects found.
* Product owner has accepted the requirement.
* When the software product is delivered to the end user.

What is Software Acceptance Criteria?

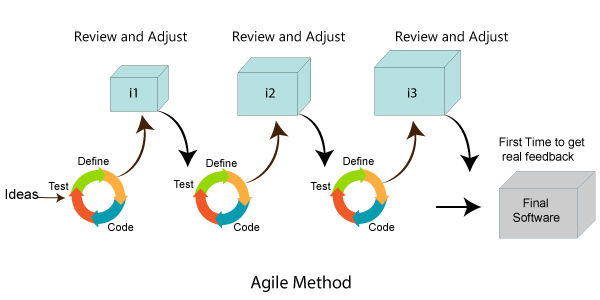
Acceptance Criteria is defined as the functionality, behavior, and performance required by a product owner. It defines what is to be done so that the developer knows when a user requirement is complete.

Advantage of Agile Methodology

There are various advantages of using agile methodology over traditional waterfall model or others. Agile development methodology and testing practices have worked wonders for numerous organizations with positive aspects. Its positive aspects are not hidden, it is very much visible in the organization.

Advantages of Agile Methodology

1. Customer satisfaction is rapid, continuous development and delivery of useful software.
2. Customer, Developer, and Product Owner interact regularly to emphasize rather than processes and tools.
3. Product is developed fast and frequently delivered (weeks rather than months.)
4. A face-to-face conversation is the best form of communication.
5. It continuously gave attention to technical excellence and good design.
6. Daily and close cooperation between business people and developers.
7. Regular adaptation to changing circumstances.
8. Even late changes in requirements are welcomed.

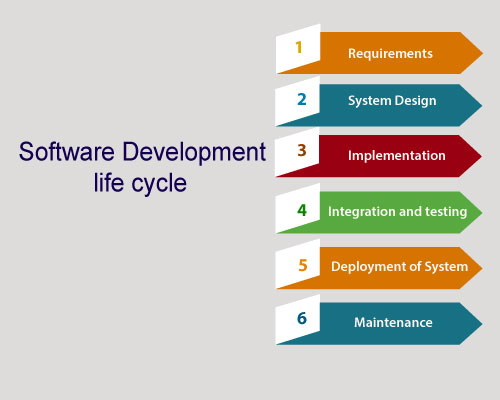


Disadvantages of Agile methodology:

1. It is not useful for small development projects.
2. There is a lack of intensity on necessary designing and documentation.
3. It requires an expert project member to take crucial decisions in the meeting.
4. Cost of Agile development methodology is slightly more as compared to other development methodology.
5. The project can quickly go out off track if the project manager is not clear about requirements and what outcome he/she wants.

Advantages of the Waterfall Model:

1. It is one of the easiest and traditional model to manage. Because of its traditional development nature, each phase has specific deliverables and a review process.
2. It works well in smaller size projects where requirements are easily understandable.
3. It has a faster product delivery model.
4. There are well-documented process and results.
5. Easily adaptable method for shifting teams
6. This project management methodology is beneficial to manage dependencies.



Disadvantages of Waterfall Model:

1. It is not an ideal model to develop a large scale project size.
2. It requires a clear-cut requirement at the beginning time; otherwise, it may lead to a less effective method.
3. It is difficult to move back to make changes in the previous phase.
4. The testing process starts once development is completed. Hence, it has high chances of bugs to be found later in project development. Due to this, it is costly to fix.

Compression between the Agile methodology and Waterfall model:

|  |  |
| --- | --- |
| **Agile methodology** | **Waterfall model** |
| It follows the incremental approach. | It is a sequential design process. |
| It divides the project development lifecycle into a sprint. | The software development process is divided into distinct phases. |
| Agile methodology is a flexible methodology. | The Waterfall is a structured software development methodology. |
| Agile is the collection of many different projects. | It is completed as one single project. |
| The test plan is reviewed after each sprint | Test plan is reviewed after complete development. |
| Testing team can take part in the requirements change phase without problems. | It is difficult for the test to initiate any change in needs. |

Agile Manifesto

In February 2001, at the Snowbird resort in Utah, a team of 17 software developers met to discuss lightweight development methods. The result of their meeting was the following Agile Manifesto for software development:-

We are uncovering the better ways of developing software by doing it and helping others to do it. Through this meeting, we have come to value -

* Individuals and interactions over Processes and tools.
* Working software over comprehensive documentation.
* Customers are collaboration over contact negotiation.
* Responding to change over following a plan.

So that, while there is value in the items on the right, we value the items on the left more.

The Twelve Principle of Agile Manifesto

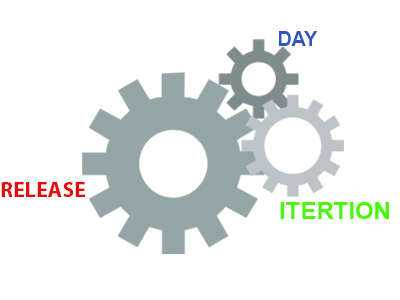
1. **Customer Satisfaction:** Manifesto provides high priority to satisfy the costumer's requirements. This is done through early and continuous delivery of valuable software.
2. **Welcome Change:** Making changes during software development is common and inevitable. Every changing requirement should be welcome, evenin the late development phase. Agile process works to increase the customers' competitive advantage.
3. **Deliver the Working Software:** Deliver the working software frequently, ranging from a few weeks to a few months with considering the shortest time period.
4. **Collaboration:** Business people (Scrum Master and Project Owner) and developers must work together during the entire life of a project development phase.
5. **Motivation:** Projects should be build around motivated team members. Provide such environment that supports individual team members and trust them. It makes them feel responsible for getting the job done thoroughly.
6. **Face-to-face Conversation:** Face-to-face conversation between Scrum Master and development team and between the Scrum Master and customers for the most efficient and effective method of conveying information to and within a development team.
7. **Measure the Progress as per the Working Software:** The working software is the key and primary measure of the progress.
8. **Maintain Constant Pace:** The aim of agile development is sustainable development. All the businesses and users should be able to maintain a constant pace with the project.
9. **Monitoring:** Pay regular attention to technical excellence and good design to maximize agility.
10. **Simplicity:** Keep things simple and use simple terms to measure the work that is not completed.
11. **Self-organized Teams:** The Agile team should be self-organized. They should not be depending heavily on other teams because the best architectures, requirements, and designs emerge from self-organized teams.
12. **Review the Work Regularly:** The work should be reviewed at regular intervals, so that the team can reflect on how to become more productive and adjust its behavior accordingly.

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.Differenceen JDK, JRE, and JVM

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.

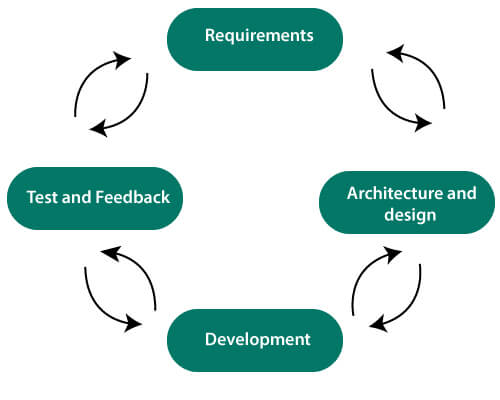
Small, Cross-functional Teams

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.

Agile Software Development Life Cycle (SDLC)

**Software development life cycle (SDLC)** is a phenomenon to **design**, **develop** and, **test** high-quality software. The primary aim of SDLC is to produce high-quality software that fulfills the customer requirement within times and cost estimates.

**Agile Software Development Life Cycle (SDLC)** is the combination of both iterative and incremental process models. It focuses on process adaptability and customer satisfaction by rapid delivery of working software product. Agile SDLC breaks down the product into small incremental builds. These builds are provided into iterations.



In the agile SDLC development process, the customer is able to see the result and understand whether he/she is satisfied with it or not. This is one of the advantages of the agile SDLC model. One of its disadvantages is the absence of defined requirements so, it is difficult to estimate the resources and development cost.

**Each iteration of agile SDLC consists of cross-functional teams working on various phases:**

1. Requirement gathering and analysis
2. Design the requirements
3. Construction/ iteration
4. Deployment
5. Testing
6. Feedback

Requirements gathering and analysis

In this phase, you must define the requirements. You should explain business opportunities and plan the time and effort needed to build the project. Based on this information, you can evaluate technical and economic feasibility.

Design the requirements

When you have identified the project, work with stakeholders to define requirements. You can use the user flow diagram or the high-level UML diagram to show the work of new features and show how it will apply to your existing system.HTML Tutorial

Construction/ Iteration

When the team defines the requirements, the work begins. The designers and developers start working on their project. The aims of designers and developers deploy the working product within the estimated time. The product will go into various stages of improvement, so it includes simple, minimal functionality.

Deployment

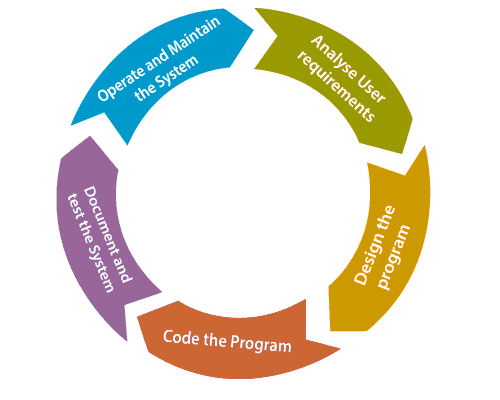
In this phase, the team issues a product for the user's work environment.

Testing

In this phase, the Quality Assurance team examine the product's performance and look for the bug.

Feedback

After releasing of the product, the last step is to feedback it. In this step, the team receives feedback about the product and works through the feedback.



Agile SDLC Process Flow

1. **Concept:** Project are imagined and prioritized.
2. **Inception:** Team members are created, funding is put in place, and basic environments and requirements are discussed.
3. **Iteration/Constriction:** The software development team works to deliver working software. It is based on requirement and feedback.
4. **Release:** Perform quality assurance (QA) testing, provides internal and external training, documentation development, and final version of iteration into the product.
5. **Production:** It is ongoing support of the software.

Advantages of Agile SDLC

1. Project is divided into short and transparent iterations.
2. It has a flexible change process.
3. It minimizes the risk of software development.
4. Quick release of the first product version.
5. The correctness of functional requirement is implemented into the development process.
6. Customer can see the result and understand whether he/she is satisfied with it or not.

Disadvantages of Agile SDLC

1. The development team should be highly professional and client-oriented.
2. New requirement may be a conflict with the existing architecture.
3. With further correction and change, there may be chances that the project will cross the expected time.
4. There may be difficult to estimate the final coast of the project due to constant iteration.
5. A defined requirement is absent.

Agile Project Management

Agile project management is an interactive approach to manage software development. The agile project management focuses on continuous releases and covers customer feedback with every iteration.

Traditionally the agile project management is classified into two frameworks: **scrum** and **kanban**. The [scrum framework](https://www.javatpoint.com/agile-scrum) focused fixed-length project iterations, whereas [kanban framework](https://www.javatpoint.com/agile-kanban) focused on continuous releases. After competition of project first iteration (or steps) project management activity immediately moves on to the next.

History of Agile Project Management

Agile project management is rapidly rising in the 21st century. It is used for software development projects and other IT initiatives.

However, from the mid-20th century, the concept of continuous development has taken various forms. For example, there was James Martin's **Rapid Iterative Production Prototyping (RIPP)**, an approach that served as the premise for the 1991 book **Rapid Application Development (RAD)**.

The agile project management framework which has emerged in most recent years is known as Scrum. This methodology features works on the development team to create a product backlog. It also creates a prioritized list of the features, functionalities, and fixes required to deliver a successful software system. The scrum team offers the pieces of a task in rapid increments.

How Agile Project Management works

The agile project management calls for teams to regularly evaluate cost and time as they move through their work. They use velocity, burnup and burndown charts to measure their work, rather than Gantt charts and project milestones to track progress.HTML Tutorial

The agile team practices to continuous development and continuous integration using technology that automates steps to speed up the release and use of products.

The presence and participation of the project manager are not required in agile project management. Although the presence of the project manager is essential for success under the traditional (waterfall model) project delivery. The role of the project manager is to distribute task among team members. However, the project manager is not obsolete in agile project management, and many organizations use them in a large, more complex project. The organization mostly places them in the project coordinator role.

Agile Project Management demands that team members know how to work in this new agile methodology. The team member must be able to coordinate with each other, as well as with users.

How to implement Agile

1.SCRUM

2. EXtreme Programming (XP)

3. LEAN

4. KANBAN

5. CRYSTAL

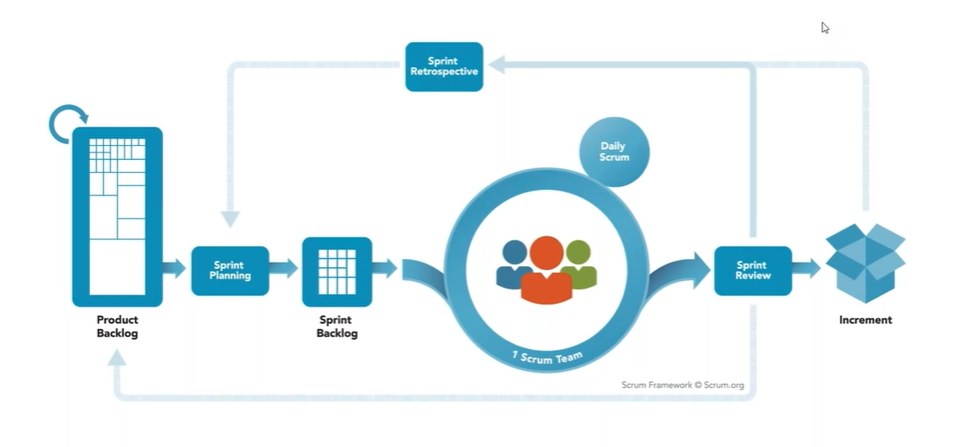
What is Scrum?

**Scrum is a framework** that helps agile teams to work together. Using it, the team members can deliver and sustain the complex product. It encourages the team to learn through practice, self-organize while working on the problem. Scum is a work done through the framework and continuously shipping values to customers.

It is the most frequent software that is used by the development team. Its principle and lessons can be applied to all kinds of teamwork. Its policy and experiences is a reason of popularity of Scrum framework. The Scrum describes a set of tools, meetings, and roles that help the teams structure. It also manages the work done by the team

The framework

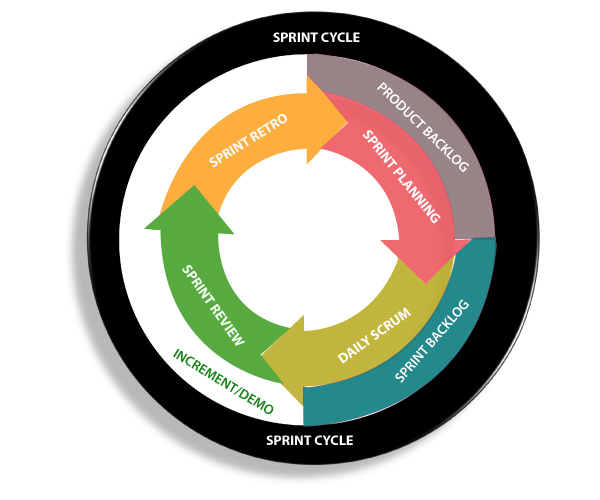
Scrum and agile are not the same thing because Scrum focused on continuous improvement, which is a core foundation of agile. Scrum framework focuses on ongoing getting work done.



What are sprints?

With scrum, a product is built in a series of repetition called **sprints**. It breaks down big complex projects into bite-size pieces. It makes projects more manageable, allows teams to ship high quality, work faster, and more frequently. The sprints give them more flexibility to adapt to the changes.

Sprints are a short, time-boxed period for Scrum team that works to complete a set amount of work. Sprints are the core component of Scrum and agile methodology. The right sprints will help our agile team to ship better software.



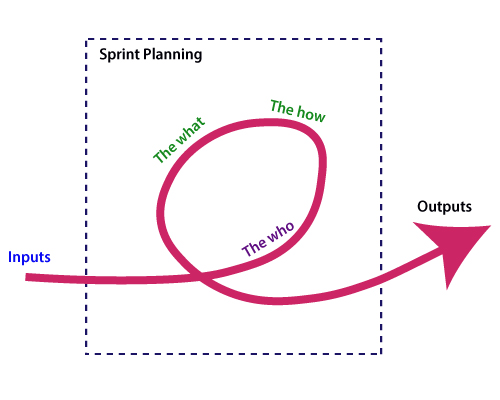
What is sprint plan?

Sprint plan is an action in Scrum that kicks off the sprint. The primary purpose of sprint plan is to define what can deliver in the sprint. It also focuses on how the work will be achieved. It is done in combination with the whole Scrum team members REATE TABLE

The sprint is a set of the period where all the work to be done. Before we start the development, we have to set up the sprint. We need to describe how long time is required to achieve the sprint goal and where we are going to start.

Factors affecting Sprint planning

* **The What:** The product owner describes the goal of the sprint and the backlog items which contribute to achieve that goal.
* **The How:** Agile development team plans its necessary work on how to achieve and deliver the sprint goal.
* **The Who:** The product owner defines the goal based on the value that the customers seek. And the developer needs to understand how they can or cannot deliver that goal.
* **The Inputs:** The product backlog provides the list of input stuff that could potentially be part of the current sprint. The team looks over the existing work done in incremental ways.
* **The Outputs:** The critical outcome of sprint planning is to meet described team goal. The product set the goal of sprint and how they will start working towards the goal.



What is the product backlog?

A product backlog is a registered list of work for the development team. It is driven from the roadmap and its requirements. The essential task is represented at the top of the product backlog so that the team member knows what to deliver first. The developer team doesn't work through the backlog from the product owner's side and product owner doesn't push the work to the developer team. The developer team pulls work from the product backlog.

Backlog starts with the two "R"s

The fundamental product backlog is provided by a team's **roadmap** and **requirements**. Roadmap repetition breaks down into several epics, and each epic will have several requirements and user stories.

The product owner organizes each of the customer stories into a single list. This story is organized for the development team. The product owner chooses to deliver first complete epic.

The factors that influence a product owner's prioritization

* Priority of customer
* Importance of getting feedback
* Relative implementation difficulty
* Symbiotic relationships between work items

# Difference between Agile and Scrum (Agile vs Scrum)

[Agile](https://www.javatpoint.com/agile-methodology) is an iterative approach of software development methodology using short iterations of 1 to 4 weeks. Due to the agile methodology, the development process is aligned to deliver the changing business requirement. Using Agile methodology, the software is distributed with faster and fewer changes.

[Scrum](https://www.javatpoint.com/agile-scrum) is a framework of agile that helps agile teams to work together. Using it, the team members development, deliver and sustain the complex product. It encourages the team to learn through practice, self-organize while working on the problem. Scum is a work done through the framework and continuously shipping values to customers.

## Agile

1. Agile is an **iterative and incremental approach** to software development methodology.
2. In this approach, the **leadership** plays an important role.
3. Agile software development is highly suitable for the **medium or large project**.
4. **Flexibility** is the most significant advantage of agile as it quickly reacts to changes.
5. Agile involves **face-to-face communication** and collaboration between the members of various cross-functional teams.
6. Agile development needs **frequent delivery** to the end user for their feedback.
7. In this development, each step like requirements, analysis, design, are **continually monitored** during the lifecycle.
8. The **project leader** takes cares of all the tasks in the agile method.
9. End-user may give their **feedback during the development** process. So, the end product will be more useful.
10. **Delivery and update** of the software are taking place regularly.
11. Design and execution should be kept **simple**.
12. The priority of agile development is always to satisfy the customer by providing **continuous delivery** of valuable software.
13. Working software is the most **fundamental measure** of progress.
14. It is best to have **face-to-face communication** to get as close to the project goal as possible.

## Scrum

1. Scrum is a framework of agile methodology. In which **incremental builds** are delivered to end user in every two to three weeks.
2. Scrum's team is **self-organized**, cross-functional team.
3. Scrum is used in the project where the requirement **rapidly** changes.
4. A compared to agile it is more **rigid**. So that there are no chances of frequent change.
5. In **daily stand up meeting** the teamwork is achieved with a fixed role assigned to team members, scrum master, and product owner.
6. **No need to change many more** while implementing scrum process.
7. In this process, a **build is delivered** after each sprint to the client for their feedback.
8. After every sprint a demonstration of functionality is provided. So that the **regular feedback** can be taken before next sprint.
9. There is no team leader, so the **entire team handles the issues** or problems.
10. When the team completes the **current sprint activity**, then the next sprint is planned.
11. Design and execution can be **innovative and experimental**.
12. The **daily sprint meeting** is organized to review the feedback to decide the future progress of the project.
13. Working software is **not a fundamental measure**.
14. The target of the Scrum team is to deliver **maximum business value.**

Agile Daily Stand-up

Agile daily stand-up is termed as the day-to-day status meeting on the project of the members of the agile team. The daily meeting of the agile team discussed the forum for regular updates as well as the problems of team members. It focuses on addressing the issues and tries to solve the issues quickly. The daily stand-up is the regular practice, no matter how an agile team is established regardless of its office location.

What is Daily Stand-up?

The daily stand-up is a daily status meeting of the agile team member. This meeting roughly takes 12 to 18 minutes (an average of 15 minutes).

Each member of the team has to answer three important questions

1. What he/she did yesterday?
2. What he/she will do today?
3. The problem he/she is facing . . . He/she blocked due to. . .

The daily stand-up is done for a day-to-day status update. The meeting of team members with the product owner can be scheduled at different time. The participants in the stand-up meetings only stand instead of sitting so that the meetings get finished quickly.

Important of Stand-up:

The importance of having a daily stand-up in agile are as follows:

* The team can evaluate the progress report daily.
* The team member discusses all the progress and the commitments he/she made for the day.
* The members can also see whether they can deliver the project as per the iteration plan or not.
* Stand-up provides visibility to the team on any delay that occurs due to some obstacles.

Who Attends a Stand-up?

* The project owner, scrum master, and the delivery team should attend the stand-up regularly.
* Customers and Stakeholders are encouraged to participate in the meeting, and they act as an observer. However, they are not supposed to participate in stand-ups.
* The responsibility of scrum master is to take note each team member's queries and the problems they are facing.

Geographically Dispersed Teams

A stand-up meeting is done in different ways depends upon the working time zone. Difference between JDK, JRE, and JVM

* On the rotation basis, select a member who can take the stand-up meeting of teams located in different time zones.
* A separate team has a separate stand-up meeting.
* Daily update the status of the stand-up in a tool such as SharePoint, Rally, Wikis, etc.
* There are varieties of communication tools ready like video conferencing, instant messengers, conference call, and other knowledge sharing tools.

Agile Definition of Done

Agile Definition of **done** is defined into three different stages called User Story (Requirement), Iteration, and product Release. These are given below:

User Story (requirement)

A user story is a requirement which is formulated into few sentences. The user requirement is the everyday language of user. This user story should be completed within iteration. The user story is done when

* All the related code and documentation have been checked-in.
* The product passed all the processes of unit test.
* All the processes of the acceptance test case have been moved.
* The product owner must have accepted the story.
* The help text (documentation) is written.

Iteration

An iteration is a time-based collection of a user story. It works on the defected product and accepted within the release of a product. Iteration is defined at the time of the iteration planning meeting and completed within the iteration demo and review meeting. The iteration is also known as a sprint. The repetition is required when:

* Performance of the product has been tested.
* Product backup is complete.
* User requirement has been accepted or moved for the next iteration.
* Defected product has been fixed or postponed for the next iteration.

Release

The product release is a major occasion that represents an internal and external delivery of work. It also tests the version of the product or system. The product release is done when:

* The system is stress tested.
* Performance is high.
* Contain the security validation in the product.
* Disaster recovery plan is tested.