# Agile Software Development

Agile software development -- also referred to simply as Agile -- is a type of development methodology that anticipates the need for flexibility and applies a level of pragmatism to the delivery of the finished product. Agile software development requires a cultural shift in many companies because it focuses on the clean delivery of individual pieces or parts of the software and not on the entire application.

Benefits of Agile include its ability to help teams in an evolving landscape while maintaining a focus on the efficient delivery of business value. The collaborative culture facilitated by Agile also improves efficiency throughout the organization as teams work together and understand their specific roles in the process. Finally, companies using Agile software development can feel confident that they are releasing a high-quality product since testing is performed throughout development, providing the opportunity to make changes as needed and alert teams to any potential issues.

Agile has replaced waterfall as the development methodology of choice in most companies, but is itself at risk of being eclipsed or consumed by the growing popularity of DevOps.

## The four values of Agile

In 2001, 17 software development professionals gathered to discuss concepts around the idea of lightweight software development and ended up creating the Agile Manifesto. The Manifesto outlines the four core values of Agile, and although there has been debate about whether the Manifesto has outlived its usefulness, it continues at the core of the Agile movement.

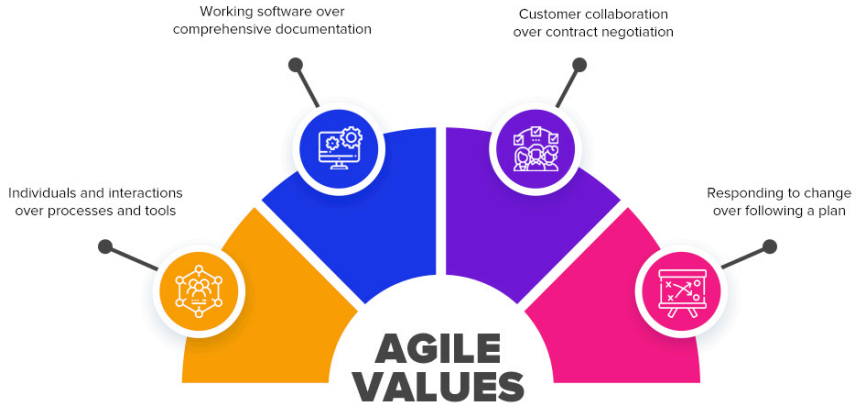
The four core values outlined in the Agile Manifesto are:

**Individual interactions are more important than processes and tools.** People drive the development process and respond to business needs. They are the most important part of development and should be valued above processes and tools. If the processes or tools drive development, then the team will be less likely to respond and adapt to change and, therefore, less likely to meet customer needs.

**A focus on working software rather than thorough documentation.** Before Agile, a large amount of time was spent on documenting the product throughout development for delivery. The list of documented requirements was lengthy and would cause long delays in the development process. While Agile does not eliminate the use of documentation, it streamlines it in a way that provides the developer with only the information that is needed to do the work -- such as user stories. The Agile Manifesto continues to place value on the process of documentation, but it places higher value on working software.

**Collaboration instead of contract negotiations.** Agile focuses on collaboration between the customer and project manager, rather than negotiations between the two, to work out the details of delivery. Collaborating with the customer means that they are included throughout the entire development process, not just at the beginning and end, thus making it easier for teams to meet the needs of their customers. For example, in Agile software development, the customer may be included at different intervals for demos of the product. However, the customer could also be present and interacting with the teams on a daily basis, attending all meetings and ensuring the product meets their desires.

**A focus on responding to change.** Traditional software development used to avoid change because it was considered an undesired expense. Agile eliminates this idea. The short iterations in the Agile cycle allow changes to easily be made, helping the team modify the process to best fit their needs rather than the other way around. Overall, Agile software development believes change is always a way to improve the project and provide additional value.




## The 12 principles of Agile

The Agile Manifesto also outlined 12 core principles for the development process. They are:

1. Satisfy customers through early and continuous delivery of valuable work.
2. Break big work down into smaller tasks that can be completed quickly.
3. Recognize that the best work emerges from self-organized teams.
4. Provide motivated individuals with the environment and support they need and trust them to get the job done.
5. Create processes that promote sustainable efforts.
6. Maintain a constant pace for completed work.
7. Welcome changing requirements, even late in a project.

8. Assemble the project team and business owners on a daily basis throughout the project.
9. Have the team reflect at regular intervals on how to become more effective, then tune and adjust behaviour accordingly.
10. Measure progress by the amount of completed work.
11. Continually seek excellence.
12. Harness change for a competitive advantage.

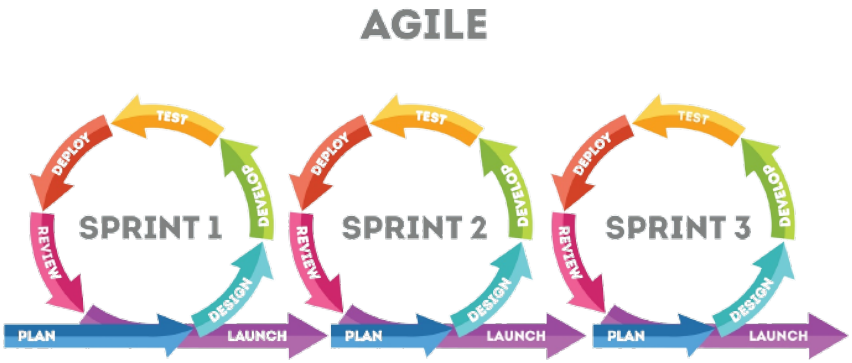
## The Agile software development cycle

The Agile software development cycle can be broken down into six steps: concept, inception, iteration/construction, release, production and retirement.

The first step, concept, involves the identification of business opportunities in each potential project as well as an estimation of the time and work that will be required to complete the project. This information can then be used to prioritize projects and discern which ones are worth pursuing based on technical and economic feasibility.

During the second step, inception, team members are identified, funding is established and the initial requirements are discussed with the customer. A timeline should also be created that outlines the various responsibilities of teams and clearly defines when work is expected to be completed for each sprint. A sprint is a set period of time during which specific work has to be completed and made ready for review.

The third step, iteration/construction, is when teams start creating working software based on requirements and continuous feedback. The Agile software development cycle relies on iterations -- or single development cycles -- that build upon each other and lead into the next step of the overall development process until the project is completed. Each iteration typically lasts between two to four weeks, with a set completion date. The goal is to have a working product to launch at the end of each iteration.




Multiple iterations occur throughout the development cycle and they each possess their own workflow. A typical iteration flow consists of:

- Defining requirements based on the product backlog, sprint backlog and customer and stakeholder feedback;
- Developing software based on the set requirements;
- Conducting quality assurance testing, internal and external training and documentation;
- Delivering and integrating the working product into production; and
- Gathering customer and stakeholder feedback on the iteration in order to define new requirements for the next sprint.

The fourth step, release, involves final quality assurance testing, resolution of any remaining defects, finalization of the system and user documentation and, at the end, release of the final iteration into production.

After the release, the fifth step, production, focuses on the ongoing support necessary to maintain the software. The development teams must keep the software running smoothly while also teaching users exactly how to use it. The production phase continues until the support has ended or the product is planned for retirement.

The final step, retirement, incorporates all end-of-life activities, such as notifying customers and final Migration. The system release must be removed from production. This is usually done when a system needs to be replaced by a new release or if the system becomes outdated, unnecessary or starts to go against the business model.

Throughout the Agile cycle, different features can be added to the product backlog, but the entire process should consist of repeating each step over and over until every item in the backlog has been satisfied. This makes the Agile cycle more of a loop than a linear process. At any time, an enterprise can have multiple projects occurring simultaneously with iterations that are logged on different product lines and a variety of internal and external customers providing different business needs.


## Types of Agile methodologies

The goal of every Agile methodology is to embrace and adapt to change while delivering working software as efficiently as possible. However, each method varies in the way it defines the steps of software development. The most widely used Agile methods include:

- Scrum
- Lean software development

- Extreme programming
- Crystal
- Kanban
- Dynamic systems development method
- Feature-driven development

Scrum is a lightweight Agile framework that can be used by project managers to control all types of iterative and incremental projects. In Scrum, the product owner creates a product backlog that allows them to work with their team to identify and prioritize system functionality. The product backlog is a list of everything that needs to be accomplished to deliver a successful, working software system -- this includes bug fixes, features and non-functional requirements. Once the product backlog is defined, no additional functionality can be added except by the corresponding team.

Once the team and the product owner have established the priorities, cross-functional teams step in and agree to deliver working increments of software during each sprint -- often within 30 days. After each sprint, the product backlog is reevaluated, analyzed and reprioritized in order to select a new set of deliverable functions for the next sprint. Scrum has gained popularity over the years since it is simple, has proven to be productive and can incorporate the various overarching practices promoted by the other Agile methods.

Lean software development is another iterative method that places a focus on using effective value stream mapping to ensure the team delivers value to the customer. It is flexible and evolving; it does not have rigid guidelines or rules. The Lean method uses the following primary principles:

- Increasing learning
- Empowering the team
- Fostering integrity
- Removing waste
- Understanding the whole
- Making decisions as late as possible
- Delivering the product as fast as possible

The Lean method relies on fast and reliable feedback between the customers and programmers in order to provide fast and efficient development workflows. To accomplish this, it provides individuals and small teams with decision-making authority instead of relying on a hierarchical flow of control. To eliminate waste, the Lean method asks users to only select truly valuable features for their system, prioritize these chosen features and then deliver them in small batches. Lean software development also encourages automated unit tests to be written simultaneously with the code and concentrates on ensuring every member of the team is as productive as possible.

The extreme programming (XP) method is a disciplined approach that focuses on speed and continuous delivery. It promotes increased customer involvement, fast feedback loops, continuous planning and testing and close teamwork. Software is delivered at frequent intervals -- usually every one to three weeks. The goal is to improve software quality and responsiveness when faced with changing customer requirements.

The XP method is based on the values of communication, feedback, simplicity and courage. Customers work closely with their development team to define and prioritize their requested user stories. However, it is up to the team to deliver the highest priority user stories in the form of working software that has been tested at each iteration. To maximize productivity, the XP method provides users with a supportive, lightweight framework that guides them and helps ensure the release of high-quality enterprise software.

Crystal is the most lightweight and adaptable methodology. It focuses on people and the interactions that occur while working on an Agile project as well as business-criticality and priority of the system under development. The Crystal method works off of the realization that every project possesses unique characteristics that require a slightly tailored set of policies, practices and processes. As a result, it is made up of a collection of Agile process models, such as Crystal Orange, Crystal Clear and Crystal Yellow. Each model has its own unique characteristics that are driven by different factors, including project priorities, team size and system criticality.

Similar to other Agile methodologies, Crystal emphasizes frequent delivery of working software with high customer involvement, adaptability and the elimination of bureaucracy and distractions. Its key principles include communication, teamwork and simplicity.

Kanban uses a highly visual workflow management method that allows teams to actively manage product creation -- emphasizing continuous delivery -- without creating more stress in the software development lifecycle (SDLC). It has become popular among teams also practicing Lean software development.

Kanban uses three basic principles: visualize the workflow; limit the amount of work in progress; and improve the flow of work. Similar to the Scrum, the Kanban method is designed to help teams work more efficiently with each other. It encourages continuous collaboration and attempts to define the best possible workflow in order to promote an environment with active and ongoing learning and improvement.

The dynamic systems development method (DSDM) is a response to the need for a common industry framework for rapid software delivery. The DSDM is based on eight key principles; failing to abide by any one of the principles introduces risk to successful completion of the project. The eight principles are:

1. Collaboration
2. On-time delivery
3. Demonstrated control

4. Continuous, clear communication
5. A constant focus on the business need
6. Iterative development
7. Creation in increments from firm foundations
8. Refusal to compromise quality

In the DSDM, rework is built into the process and all changes must be reversible. System requirements are prioritized using MoSCoW Rules, which ranks priority as:

- M -- must have
- S -- should have
- C -- could have, but not critical
- W -- won't have now, but could have later

It is important to the DSDM that not every requirement is considered critical. Each iteration should include less critical items which can be removed so higher priority requirements are not impacted.

Finally, feature-driven development (FDD) blends software engineering best practices -- such as developing by feature, code ownership and domain object modeling -- to create a cohesive, model-driven, short-iteration process. FFD begins by defining an overall model shape, which in turn creates a feature list. The method then proceeds with iterations that last two weeks and focus on planning by feature, designing by feature and building by feature. If a feature takes more than two weeks to build, then it should be broken down into smaller features. The primary advantage of FDD is that it is scalable -- even to large teams -- since it uses the concept of "just enough design initially," or JEDI.

# Roles in Agile Methodology

An agile software development process always starts by defining the users for a particular product and documenting a vision statement for the scope of problems, opportunities, and values to be addressed. The product owner captures this vision and works with a multidisciplinary team (or teams) to deliver on it. Several roles are involved in an agile development process.

## Users

An agile process always begins with the user or customer in mind. Today, we often define user personas to illustrate different workflow roles or types of customer needs and behaviours.

## Product owner

The product owner is tasked to be the voice of the customer, including any internal stakeholders. This person distils insights, ideas, and feedback to create a product vision. Product visions are often short and straightforward, but they nonetheless paint a picture of who the customer or user is, what values are being addressed, and a strategy for addressing them. I imagine Google's original vision looked something like, "Let's make it easy for anyone with internet access to find relevant websites and web pages with a simple, keyword-driven interface and an algorithm that ranks reputable sources higher in the search results." Whatever the vision, the product owner is responsible for defining it and then working with the development team to make it real.

To work with the development team, the product owner breaks down the product vision into a series of user stories. Each user story should identify the target users, their challenges, why the solution is needed, and what constraints and acceptance criteria define the solution. The product owner prioritizes these user stories and reviews them with the team to ensure they have a shared understanding of what is being asked of them.

## Scrum Team

The Scrum Team has a greater sense of pride and accomplishment and is better positioned to demonstrate their work to the group. They are allowed time to the end-user, and, in the best-case scenario, meet with the end-users face-to-face time.

Knowing the end-user is the best way to develop quality software that will offer the highest value to the end-user. By learning to accept feedback at each sprint session, the team gains more appreciation for the product owners and customers.

The team will also feel less like they are being judged or graded by the product owner. They will come away from the meeting feeling more energized and motivated towards the next sprint.

## Company Executives/Stakeholders

When executives and stakeholders are part of the review meeting, they can see the teamwork while participating in a real review of the work done and actions taken. Additionally, the end-users can personally inform the executives/stakeholders of their needs.

Most importantly, they learn to act appropriately. The dynamics of these meetings are significantly different when customers are part of the session.

When end-users are present, executives and stakeholders spend less time bickering and criticizing others. Instead, everyone is on their best behaviour, ensuring the review is more motivated and productive.

## End Users/Customers/Partners

Without question, the end-users are the stars of the review meetings because they spend valuable face-to-face time with the software developers. They spend so much time with the developing group, the end-users feel a sense of belonging.

Because they spend so much time and involvement in the development of the project, the users become the biggest company supporters. This acquired knowledge is valuable in reducing in a flood of support calls that are common after a release.

## Scrum Master

The Scrum Master facilitates the Sprint Review Meeting. This person establishes an environment among all members so that great accomplishments and goal achievements can happen. To make it happen, the Scrum Master must make sure the right people are at the meeting. He/she must also ensure the right room and supplies are readily available.

Other smaller details like ordering food and drinks are also the responsibility of the Scrum Master. While this may seem unimportant to some, food is the fuel that teams need to continue their hard work during the meetings. Personally, I have had many great conversations with teams while eating together.

One thing to keep in mind, this is not the Scrum Master's show – they are not the star. The Scrum Master is there to support the team. They also welcome the end-users and make sure they are included in the team, have the chance to be heard, and are behind the scenes.

## Software development team

Teams should be multidisciplinary and include a diverse group with the skills and backgrounds to get the job done. In addition to developers, agile development teams should include quality assurance automation engineers, data engineers, user experience (UX) designers, and other roles depending on the type of software project.

Agile focuses teams on delivering working software, so they must complete end-to-end functioning applications, integrations, and other deliverables that impact users—not just the technical components. Team members must align on what they are building, who is doing what, and how the software will be developed.

Agile teams often have other roles assigned, including the following:

- **Tech or team leads** partner with the product owner on architecture, non-functional acceptance criteria, sequencing, dependencies, and other technology and security considerations. Tech leads have broad responsibilities that might include estimating stories and planning implementation details with the team.
- **Scrum masters** often coach new teams on agile processes, responsibilities, and tools. Scrum master responsibilities can include resolving b+locks that impede progress, reviewing approaches to improve the agile team's velocity, and grooming backlogs.
- **Business analysts** partner with the product owner. The analysts' responsibilities often include creating wireframes, documenting user stories, and reviewing test results. Business analysts are especially helpful when software development teams are developing micro-services and other technical products, and where the business analyst has more software development knowledge than the product owner.

It is up to organizational leaders to decide how to staff agile teams and how big to make them. Many follow Jeff Bezos's best practice of constructing two pizza-size agile teams to maximize the collaboration between teammates.

## Advantages and disadvantages of Agile

Much has been compared over the years with Agile versus Waterfall approaches.

In the Waterfall era of software development, coders worked alone, with little to no input before handing the software to testers and then on to production. Bugs, complications and feature changes either weren't handled well, or were dealt with so late in the process that projects were seriously delayed or even scrapped.

The idea behind the Agile model, in which everyone -- including the business side -- stayed involved and informed in the development process, represented a profound change in both company culture and the ability to get better software to market more quickly.

Collaboration and communication became as important as technology, and because the Agile Manifesto is open to interpretation, Agile has been adapted and modified to fit organizations of all sizes and types. The Agile cultural shift also paved the way for the latest software development evolution, DevOps.

On the other hand, many would say the biggest disadvantage of Agile is the fact it has been modified -- some would say diluted -- by many organizations. This phenomenon is so widespread that the "Agile my way" practitioners are known as "ScrumButs," as in, "We do Scrum in our organization, but ...."

Although Agile opens up the lines of communication between developers and the business side, it's been less successful bringing testing and operations into that mix -- an omission that may have helped the idea of DevOps gain traction.

Another potential concern about Agile is its lack of emphasis on technology, which can make it difficult to sell the concept to upper managers who don't understand the role that culture plays in software development. Furthermore, the necessity of completing sprints on time can create a stressful work environment for software developers. They may be forced to work extra hours and stay late in order to meet deadlines.

## What is a Product Backlog?

Before a project begins, the Product Owner creates a list of features to add to the project. Software development teams call this list the Product Backlog. The Product Backlog should break down the tasks necessary for each item on the list. The Product Backlog should also give teams a place to document how much time the tasks on the Backlog might take.

The Product Backlog gives teams a place to understand the dimensions of the entire project. It helps them visualize how they will tackle the tasks ahead of them. It helps the Product Owner to organize and prioritize their thoughts and wishes. The team uses the dynamic Product Backlog to keep track of problems to fix in future Sprints.

The Product Backlog, as an element of Agile, is not set. Flexibility and change must play a key role in any Product Backlog. Bugs or errors might add items to the Backlog. Software development teams will drop completed items from the Product Backlog.

The Product Backlog is a living document. The Product Backlog likely grows during certain parts of the software development project. It should begin to shrink as the project nears completion.

The Product Backlog usually contains at least some of the following items:

- Bugs – Problems testers or team members have flagged.
- User stories – features needed in the software development project. User stories should be explained in plain language from the end user's perspective.
- Tasks – general work the Scrum team needs to complete.

The Product Backlog contains all the items in the software development project. The Sprint Backlog contains only the items of the Backlog specific to the current Sprint. Sprint Backlogs are the songs. The complete Product Backlog is the album.

## What is a Sprint Backlog?

Sprint Backlogs give software development teams a way to chip away a long list of items. Teams use the Sprint Backlog in a very distinct way. They do not make changes to the Sprint Backlog. They set each Sprint Backlog at the Sprint Planning Meeting. Once they set the Sprint Backlog it is set. If new issues come up during the Sprint the team adds them to the Product Backlog. They work on these issues in a later Sprint.

A Sprint Backlog proves simpler, smaller and easier to understand than the Product Backlog. Teams still need to strategize and coordinate with the Project Owner and Scrum Master to make the Sprint run smoothly. They need to understand the limits of their resources.

A well-groomed Sprint Backlog will allow the teams the right amount of time to complete their work. They should not have to rush or work crazy hours to deliver. Before a Scrum Master moves a task from the Product Backlog to the Sprint Backlog, they should plan with the development team (and possibly the Product Owner) to ensure they have capacity. Scrum Masters, however, do not own decision-making. Scrum Masters facilitate and help the team.

## differences and similarities between Product Backlog and Sprint Backlog:

| Product Backlog | Sprint Backlog |
|---|---|
| Contains all tasks for the development project | Contains only the items to complete in the current Sprint |
| Created by the Product Owner | The development team and Scrum Master create the list |
| Flexible, a living document that will change over time | Once set in the Sprint planning meeting it does not change |
| Independent parent list | Dependent on the Product Backlog |
| Specific to the project goal | Specific to the Sprint |
| Remains until the project is complete | Ends when the Sprint ends |
| Product Owner manages | Scrum team manages |

## What is Daily Stand-up?

- A daily stand-up is a daily status meeting among all team members and it is held roughly for 15 minutes.
- Every member has to answer three important questions −
  - What I did yesterday?
  - What I'll do today?
  - Any impediment I am facing.../ I am blocked due to...
- Daily stand-up is for status update, not for any discussion. For discussion, team members should schedule another meeting at a different time.
- Participants usually stand instead of sitting so that the meeting gets over quickly.

## Why Stand-up is Important?

The benefits of having a daily stand-up in agile are as follows −

- The team can evaluate the progress on a daily basis and see if they can deliver as per the iteration plan.
- Each team member informs all about his/ her commitments for the day.
- It provides visibility to the team on any delay or obstacles.

## Who Attends a Stand-up?

- The scrum master, the product owner, and the delivery team should attend the stand-up on a daily basis.
- Stakeholders and Customers are encouraged to attend the meeting and they can act as an observer, but they are not supposed to participate in stand-ups.
- It is the scrum master's responsibility to take note of each team member's queries and the problems they are facing.

## Geographically Dispersed Teams

Stand-ups can be done in multiple ways, in case the agile team members are operating from different time zones −

- Select a member on a rotational basis, who can attend the stand-up meeting of teams located in different time zones.
- Have a separate stand-up per team, update the status of the stand-up in a tool such as Rally, SharePoint, Wikis, etc.
- Have a wide variety of communication tools ready like conference call, video conferencing, instant messengers, or any other third-party knowledge sharing tools.

## WHAT IS SPRINT PLANNING?

Sprint planning is a timeboxed working session that lasts roughly 1 hour for every week of a sprint. In sprint planning, the entire team agrees to complete a set of product backlog items. This agreement defines the sprint backlog and is based on the team's velocity or capacity and the length of the sprint.

## WHO DOES IT?

Sprint planning is a collaborative effort involving a ScrumMaster, who facilitates the meeting, a Product Owner, who clarifies the details of the product backlog items and their respective acceptance criteria, and the Entire Agile Team, who define the work and effort necessary to meet their sprint commitment.

## HOW DO WE PREPARE?

Ensure all sprint candidates meet the team's definition of *ready*. In the days and weeks leading up to sprint planning, the Product Owner identify the items with the greatest value and works towards getting them to a ready state.

- Assign a relative story point value
- Remove dependencies
- Create testable examples
- Define acceptance criteria
- Meets INVEST criteria

## WHAT IS THE BACKLOG?

The product backlog can address just about anything, to include new functionality, bugs, and risks. Product backlog items (PBI's) must be small enough to complete during a sprint and should be small enough to complete within a few days. All stories must be verified that they are implemented to the satisfaction of the Product Owner.

## ENSURE RIGHT SIZING BACKLOG ITEMS

Based on historical data of the team, first determine if product backlog items are too large to complete in a sprint. In these cases, do not consider these stories as valid sprint backlog candidates. Rather, in order to consider for sprint planning, split the stories into smaller pieces. Additionally, each story must be able to stand on its own as a vertical slice. Therefore, stories should not be incomplete or process-based as a horizontal slice.

## CALCULATING A COMMITMENT

To calculate a commitment, mature teams may use a combination of both team availability and velocity. However, new teams may not know their velocity or they may not be stable enough to use velocity as a basis for sprint planning. In these cases, new teams may need to make forecasts based solely on the their capacity.

## DETERMINING VELOCITY

First of all, as velocity is unique to every team, never use another team's velocity to plan your sprint. Derive team velocity by summing the story point estimates of all completed and accepted work from the previous sprint. By tracking team velocity over time, teams will begin to focus less on utilization and consequently more on throughput.

## DETERMINING CAPACITY

For teams without a stable velocity, each team member should provide three simple measures to determine capacity. First, what are the number of ideal hours in their work day? Second, how many days in the sprint will that person be available? Third, what percentage of time will that person dedicate to this team?

## THE PLANNING STEPS

1. Remind the team of the big picture or goal
2. Discuss any new information that may impact the plan
3. Present the velocity to be used for this release
4. Confirm team capacity
5. Confirm any currently known issues and concerns and record as appropriate
6. Review the definition of *DONE* and make any appropriate updates based on technology, skill, or team member changes since the last sprint
7. Present proposed product backlog items to consider for the sprint backlog
8. Determine the needs, sign up for work, and estimate the work owned
9. Product Owner answers clarifying questions and elaborates acceptance criteria
10. Confirm any new issues and concerns raised during meeting and record
11. Confirm any assumptions or dependencies discovered during planning and record
12. Scrum Master calls for a group consensus on the plan
13. Team and Product Owner signal if this is the best plan they can make given what they know right now
14. Get back to work

## Sprint Review Meeting

In a Sprint Review meeting, the Scrum Team and stakeholders discuss the work done during the Sprint. After assessing the work done along with changes made to the Product Backlog, the group collaborates on what steps should be taken next to optimize the value of the software. These meetings are always intentionally informal to keep the group focused on encouraging feedback and collaboration.

For month-long Sprints, review meetings should last a maximum of four hours; referred to as time-boxed sessions. If the sprint sessions are shorter in duration, the review meetings will also be shorter.

The Scrum Master is responsible for scheduling the meeting and informing everyone attending of the purpose of the review. The Scrum Master is also responsible for ensuring the meeting stays within the time-box.

## The sprint review meeting should include the following:

- Attendance and participation of the Scrum Team, product owner, and invited key stakeholders.
- The Product Owner should report the items in the Product Backlog; what backlog items have been done and what have not.
- The development team discusses what went well and the problems they experienced. They should also inform the group what they did to resolve the problems.
- The development team demonstrates their completed work while answering questions about their increment.
- The product owner leads the discussion on the Product Backlog as it currently stands. They set projected completion dates based on the progress of the Sprint session.
- To give valuable input to the Sprint planning, the entire group establishes the next steps during the Sprint review meeting.
- This is a time to review potential changes in the marketplace, the valuation of the project and what areas are considering to be the most valuable. The next steps should also be outlined.
- Review the timeline, budget, potential capabilities, and marketplace to determine the next anticipated product release.

By the end of the Sprint Review Meeting, revisions should be made to the Product Backlog to better define probable backlog items for the next Sprint session. The Product Backlog can be adjusted completely to introduce new opportunities.

## What is a sprint retrospective?

The sprint retrospective is a recurring meeting held at the end of a sprint used to discuss what went well during the previous sprint cycle and what can be improved for the next sprint. The Agile sprint retrospective is an essential part of the Scrum framework for developing, delivering, and managing complex projects. Since the early 1990s, Scrum has been used to develop:

- Software
- Hardware
- Embedded software
- Networks of interacting function
- Autonomous vehicles
- And much more

Continuous, iterative improvement is a core goal in Scrum product management, and the Scrum retrospective meeting is an official opportunity to achieve this at defined, consistent intervals—when teams have wrapped up a sprint and have a space to reflect and improve how things are done. That said, continuous improvement requires continuous effort. So, it's important for teams to consciously apply the lessons learned in each retrospective to upcoming sprints.

## What is the purpose of a sprint retrospective?

The Scrum sprint retrospective is a time boxed meeting that takes place after the sprint review and before sprint planning. Its purpose is to:

- Examine how the just-completed sprint went as far as people, relationships, processes, and tools.
- Identify and order what went well.
- Do the same with things that didn't go well.
- Identify potential improvements.
- Create a plan for implementing improvements to the way the Scrum team accomplishes its work.

Everything that affects how the Scrum team develops the product is open to discussion and improvement, including processes, tools, artifacts, environment, and so on. It allows development teams to adapt Scrum to their particular circumstances.

Scheduling a Scrum retrospective at the end of every sprint ensures that needed changes are understood and implemented before they are lost in the rush of new work. It helps each Scrum team member to identify how they can improve the specific things they contributed to the sprint, asking:

- What work has been done well in this sprint?
- What work hasn't been done well?
- What should we start doing to improve?

During each Agile sprint retrospective, the development team focuses on increasing product quality by improving work processes or adapting the definition of "done." This definition may vary from Scrum team to Scrum team. But the whole team must have the same understanding of "done" to assess when work has met expected standards.

As Scrum teams gain more experience, their definition of "done" will evolve, including more demanding criteria for higher-quality results.

## Who runs a sprint retrospective meeting?

The sprint retrospective team usually includes all development team members, the Scrum Master, and the product owner. The development team is everyone who is designing, building, and testing the product. Collectively, its members have a wide range of valuable perspectives for identifying process improvements from different points of view.

The Scrum Master facilitates the meeting and encourages the development team to improve its workflow practices within the Scrum process framework, so improvements can be enacted during the next sprint. The Scrum master is also the process coach for the Scrum team, pointing out when it is not adhering to an agreed-upon method of proceeding and providing ideas and expertise as needed.

Stakeholders and managers who are not directly part of the team usually don't come to a Scrum retrospective unless specifically invited. They participate in the Sprint review meeting instead, where the Scrum team shows what they accomplished during the sprint, often including product demos.

# How to run a sprint retrospective:

There are many ways to run a sprint retrospective. One of the most common is using a start-stop-continue approach. Each development team member is asked to identify the things the team should start doing, the ones they should stop doing, and the things they should continue doing.

The Scrum Master can facilitate this process by asking attendees to call out ideas during the Scrum, or they can go around the room and get feedback on what to start, stop, and continue in a more orderly fashion, person by person.

# Agenda

While the agendas for sprint retrospective meetings can vary, they generally cover these common steps:

- **Setting the goal**—Establish the objectives of the meeting up front, such as aiming to improve daily Scrum stand-ups, enhance communication with stakeholders or product owners, change operating rules, or something else.
- **Gathering essential data**—Draw on everyone's experience and perspective to create a shared body of information.
- **Developing insights**—From the amassed data, identify useful patterns and see the big picture, always asking why things happened the way they did.
- **Deciding on the next steps**—Identify the issues and challenges the team will tackle, and put in place a concrete plan of how to achieve success for each one.
- **Closing the retrospective**—Clarify and summarize the meeting, thank participants, and consider how future retrospectives could be improved.

# Length

The length of the Scrum retrospective meeting can vary according to:

- How many team members there are
- Whether the team is distributed and needs to meet remotely
- How many new team members have to be brought up to speed

That said, as a rule of thumb, sprint retrospectives often run:

- 45 minutes for a one-week sprint
- 1.5 hours for a two-week sprint
- 2.25 hours for a three-week sprint
- 3 hours for a month-long sprint

# Agile Scrum Interview Questions

**1. What is the duration of a scrum sprint?**
Answer: Generally, the duration of a scrum sprint (scrum cycle) depends upon the size of project and team working on it. The team size may vary from 3-9 members. In general, a scrum script complete in 3-4 weeks. Thus, on an average, the duration of a scrum sprint (scrum cycle) is 4 weeks. This type of sprint-based Agile scrum interview questions is very common in an agile or scrum master interview.

**2. What is Velocity?**
Answer: Velocity question is generally posed to understand if you have done some real work and familiar with the term. Its definition "Velocity is the rate at which team progresses print by sprint" should be enough. You can also add saying the important feature of velocity that it can't be compared to two different scrum teams.

**3. What do you know about impediments in Scrum? Give some examples of impediments.**
Answer: Impediments are the obstacles or issues faced by scrum team which slow down their speed of work. If something is trying to block the scrum team from their getting work "Done" then it is an impediment. Impediments can come in any form. Some of the impediments are given as –

- Resource missing or sick team member
- Technical, operational, organizational problems

- Lack of management supportive system
- Business problems
- External issues such as weather, war etc
- Lack of skill or knowledge
- While answering impediments related agile scrum interview questions remember that you may be asked the way to remove any of the mentioned impediment.

### 4. What is the difference and similarity between Agile and Scrum?

Answer: Difference between Agile and Scrum – Agile is a broad spectrum, it is a methodology used for project management while Scrum is just a form of the Agile that describes the process and its steps more concisely. Agile is a practice whereas scrum is a procedure to pursue this practice.

The similarity between Agile and Scrum – The Agile involves completing projects in steps or incrementally. The Agile methodology is considered to be iterative in nature. Being a form of Agile, Scrum is same as that of the Agile. It is also incremental and iterative.

### 5. What is the increment? Explain.

PMI ACP Free Test

Answer: This is one of the commonly asked agile scrum interview questions and a quick answer can be given this way. An increment is the total of all the product backlogs items completed during a sprint. Each increment includes all the previous sprint increment values as it is cumulative. It must be in the available mode in the subsequent release as it is a step to reach your goal.

### 6. What is the "build-breaker"?

Answer: The build-breaker is a situation that arises when there is a bug in the software. Due to this sudden unexpected bug, compilation process stops or execution fails or a warning is generated. The responsibility of the tester is then to get the software back to the normal working stage removing the bug.

### 7. What do you understand by Daily Stand-Up?

Answer: You may surely get an interview question about daily stand-up. So, what should be the answer to this question? The daily stand-up is an everyday meeting (most preferably held in the morning) in which the whole team meets for almost 15 minutes to find answer to the following three questions –

- What was done yesterday?
- What is your plan for today?
- Is there any impediment or block that restricts you from completing your task?
- The daily stand-up is an effective way to motivate the team and make them set a goal for the day.

### 8. What do you know about Scrum ban?

Answer: Scrum-ban is a Scrum and Kanban-based model for the software development. This model is specifically used for the projects that need continuous maintenance, have various programming errors or have some sudden changes. This model promotes the completion of a project in minimum time for a programming error or user story.

### 9. State some of the Agile quality strategies.

Answer: Some of the Agile quality strategies are –

- Iteration
- Re-factoring
- Dynamic code analysis
- Short feedback cycles
- Reviews and inspection
- Standards and guidelines
- Milestone reviews

### 10. Do you know about Agile Manifesto & its Principles? Explain in brief.

Answer: This is the theory which most of agile/scrum roles aspirant should be on tips. Four manifesto values and 12 principles should be explained as much as possible as part of this question. Even if it's not explained in 100% accurate manner it should be fine but intentions of values and principles should come out e.g.

- Working Software should be demonstrated at regular intervals
- Individuals & interaction – self-organization, self-motivating should be encouraged
- Customer collaboration
- Welcoming change at any point in time in the project

### 11. Is there any drawback of the Agile model? If yes, explain.

Answer: Yes, there are some drawbacks of the Agile model, some of them are as follows –

It is not easy to make a prediction about the effort required to complete a task. It becomes more problematic in case of large projects as it becomes difficult to get an idea of the total effort required.

At sometimes, it's not possible to properly focus on the design and documentation of the project

In case the requirements of the client are not understood properly, the final project will not meet the customer requirements. Thus, it will lead to the customer dissatisfaction.

Only the leader who has considerable experience in Agile methodologies is capable to take important decisions. The team members with little or no experience are not involved in decision-making, thus they don't get chance to advance their knowledge.

It's not always the case that you will be asked the questions about the characteristics and advantages of the agile and scrum in an agile scrum interview. So, just prepare yourself for the drawbacks and disadvantages related agile scrum interview questions.

### 12. What is the use of burn-up and burn-down charts?

Answer: The burn-up chart illustrates the amount of completed work in a project whereas the burn-down chart depicts the amount of work remained to complete a project. Thus, the burn-up and burn-down charts are used to trace the progress of a project.

### 13. Define Zero Sprint and Spike in Agile.

Answer: To answer this question, describe Zero Sprint and Agile in detail, as follows –
Zero sprint – Zero Sprint can be defined as the preparation step of the first sprint in Agile. There are some activities that are required to be done before actually starting the project. These activities are considered as the Zero sprint; the examples of such activities are – setting the environment for development, preparation of backlogs etc.
Spike – Spike is the type of story that can be taken between the sprints. Spikes are commonly used for the activities related to the design or technical issues such as research, design, prototyping, and exploration. There are two types of spikes – functional spikes and technical spikes.

### 14. What is the role of the Scrum Master?

Answer: Here's how you can answer Scrum Master interview questions like this –
The scrum master is the leader as well as coach of the Scrum team. The scrum master is responsible to serve and protect his team from any kind of distractions that could affect their performance. The main role of the scrum master is to motivate his team to achieve the sprint goal. He is focused to build a self-organized and motivated team where each member is familiar with the implementation of Agile and Scrum principles and applications. The scrum master keeps a proper check on the scrum team if they are executing committed tasks properly. He is also responsible to increase the efficiency and productivity of the team so that they can achieve the sprint goal effectively.

### 15. What do you know about a story point in Scrum?

Answer: A story point in Scrum is the unit for the estimation of total efforts that are required to perform or complete a particular task. So, here is how you can answer such agile scrum interview questions on a single line.

### 16. What is the role of Sashimi in Scrum methodology?

Answer: Sashimi plays an important role in Scrum methodology. Sashimi is a technique used by Scrum to check the completion of all the functions created by the developers. Using this technique, all the requirements such as analysis, designing, coding, testing and documentation that are used in the constitution of a product are checked and only after that the product is displayed.

### 17. What are the different roles in Scrum?

The three scrum roles i.e. Scrum Master, Product Owner and Team should be explained with the details of few primary responsibilities of each role. You can add more details as mentioned below for a particular depending on the role you are getting interviewed for.
Product owner – A product owner is actually the stakeholder of the project. He represents the project requirements before the team. He is responsible to have a vision of what to build and convey his detailed vision to the team. He is the starting point of an agile scrum software development project.
Scrum team – Scrum team is formed by the collective contribution of individuals who perform for the accomplishment of a particular project. The team is bound to work for the timely delivery of the requested product.
Scrum master – Scrum master is the leader and the coach for the scrum team who checks whether the scrum team is executing committed tasks properly. He is also responsible to increase the efficiency and productivity of the team so that they can achieve the sprint goal effectively.

### 18. What are the responsibilities of a Scrum Master?

Answer: Key responsibilities of a Scrum Master involves:

- Tracking and monitoring
- Understanding requirements properly
- Work to reach the project goal
- Process checking master and quality master
- Protect the team from detachments
- Improving the performance of the team
- Lead the meetings and resolve issues
- Resolution of conflicts and impediments
- Communication and reporting

### 19. What are different ceremonies and their importance in Scrum?

Scrum planning, Scrum – Daily stand up, Scrum review & scrum retrospective ceremonies should be clearly expressed with the purpose of the ceremony. It's important to remember the time-boxing of the ceremonies for a standard 4 weeks of Sprint or as per the Sprint you have used in your projects.

### 20. What do you understand by the term Agile testing?

Answer: Agile testing is a software testing practice that is fully based on the agile principles of software development. It is an iterative methodology where the requirements are the outcome of collaboration between the product owner and team. The agile principles and applications are applied to meet the customer requirements by successful completion of the project.

### 21. State some major principles of Agile testing.

Answer: Some major principles of Agile testing are –

- Customer satisfaction
- Face to face communication
- Sustainable development
- Quick respond to changes
- Continuous feedback
- Successive improvement
- Self-organized
- Focus on essence
- Error-free clean node
- Collective work

### 22. What are the skills of a good Agile Tester?

Answer: An agile tester is one who implements agile software development principles for software testing. Followings are the skills of a good agile tester –

- Required to be familiar with the concepts and principles of Agile
- Should have an excellent communication to communicate with the team and the clients
- Ability to set priority for the tasks according to the requirements
- Should be able to understand the requirements properly
- Understanding of the risks involved with a project due to changing requirements

### 23. What do you understand by the term "Scrum of Scrums"?

Answer: This is one of the commonly asked scrum master interview questions. Consider a case to understand the meaning of the term scrum of the scrums. Let us assume an active project on which seven teams are currently working. The number of members of each team is also seven. Each team is responsible to lead its own scrum meeting. But, in order to coordinate and communicate with different teams, it is required to organize a separate scrum meeting. The scrum meeting organized to hold a coordination between scrum teams is known as the scrum of scrums. There is one team leader from every team, known as ambassador, who is responsible to represent his team in the scrum of scrums.

### 24. Scrum is an Agile framework, right? Name a few other Agile frameworks.

Answer: Yes, Scrum is an Agile framework. Few other Agile frameworks are –
Feature Driven Development
Test Driven Development
Kanban
While answering this type of Agile Scrum interview questions please note that name the frameworks you are familiar with or have followed.

### 25. Explain some common metrics for Agile.

Answer: You may definitely come across agile scrum interview questions regarding agile metrics. The question may be related to a particular agile matric or explaining all the metrics. So, the detailed description of some common metrics for Agile is as follows:
**Velocity** – Velocity is the average number of points from last 3-4 sprints. It is measured by the summation of the all approved estimates of the stories. It gives an idea of the capacity, progress etc.
**Cumulative Flow Diagram** – With the help of a cumulative flow diagram, an inspection is done over the uniform workflow. In this diagram/graph, the x-axis represents time whereas the y-axis represents the number of efforts.
**Work Category Allocation** – Work category allocation is an important factor that gives a quick information of the time investment i.e. where the time is being invested and which task should be given priority as a factor of time.
**Time Coverage** – It is the time that is given to a code during testing. It is calculated in percentage as a factor of the number of lines of code called by the test suite and the total number of relative lines of code.
**Business Value Delivered** – It is a term which denotes the working efficiency of the team. The business objectives are assigned numerical values 1,2,3.. and so on, as per the level of priority, complexity, and ROI.
**Defect Removal Awareness** – It is the factor that helps the team to deliver a quality product. The identification of an active number of defects, their awareness, and **removal** plays an important role in delivering a high-quality product.
**Defect Resolution Time** – It is a procedure through which the team members detect the defects (bugs) and set a priority for the defect resolution. The procedure of fixing errors/bugs or defect resolution comprises of multiple processes such as clearing the picture of defect, schedule defect fixation, completing defect fixation, generation, and handling of resolution report.
**Sprint Burn Down Matric** – The sprint burndown chart is a graph to represent the number of non-implemented or implemented sprints during as Scrum cycle. This matric helps to track the work completed with the sprint.

### 26. Is it ever suggested to use waterfall over Scrum? If yes, explain when.

Yes, sometimes it is suggested to use a waterfall model over Scrum. It is done when the customer requirements are simple, well-defined, fully understood, predictable, and are not subjected to change until the completion of the project. It may the case that you would haven't ever used waterfall over Scrum but you need to prepare for such Agile Scrum interview questions.

### 27. Why does Scrum encourage the use of automated testing for projects?

Answer: Scrum encourages the use of automated (automated performance or automated regression) testing to make the fastest possible delivery of the project. While answering this question, you may explain some tools that you have used for automated testing.

**28. What do you know about "Planning Poker" technique?**
Answer: Planning poker, also known as Scrum Poker, is a card-based agile technique that is used for planning and estimation. To start a session of planning poker technique, the agile user story is read by the product owner. The steps performed in the poker planning technique are –
Each estimator has a deck of poker cards with the values such as 0, 1, 2, 3, 5, and so on, to denote story points, ideal days or something else that the team uses for estimation.
Each estimator has a discussion with the product owner and then privately selects a card on the basis of their independent estimation.
If the cards with same value are selected by all estimators, it is considered as an estimate. If not, the estimator discusses the high and low value of their estimates.
Then again, each estimator privately selects a card and reveals. This process of poker planning is repeated to reach a general agreement.

**29. Name some methodologies and development where you have used the Agile model.**
Answer: While answering this type of agile scrum interview questions, keep in mind to mention those methodologies that are familiar with. Some of the methodologies and development where the Agile model can be used are –

- Crystal methodologies
- Lean software development
- Dynamic development
- Feature-driven development

**30. Share your experience as a Scrum Master/Product Owner/Agile team member and what were your primary responsibilities?**
Answer: Here you have to explain your project details where you worked in Scrum team and defining your role with the responsibilities you held. The trick in this question is whether while explaining you are showing self-organizing and self-motivational team. Also, the interviewer will try to judge how in depth you have worked in the agile/scrum environment based on your explanation.

**31. What was the length of sprints/iterations in your project?**
Answer: This is probably the most common question asked in agile interviews. The idea here is to judge in which kind of environment you have worked. There will be definitely follow up question like was this length fixed in the beginning and never changed? Have you tried with more than this length or less than that?

**32. How have you done user story mapping & estimation of stories in your projects?**
Answer: This question is to under how you have done story writing, mapping, and estimation. Have you used any estimation technique like planning poker, t-shirt, sizing etc? Whatever technique you used in your project just mention it very clearly.

**33. What is the biggest challenge you faced in your project while handling the Scrum team members?**
Answer: This question is pure to judge your experience. The better you articulate your challenges working in agile better it will be. Challenges generally faced in the initial stages of scrum is stabilizing the velocity, team members conflicts, sticking to time-boxing etc..

**34. Have you ever performed the removal of impediments as a scrum master on behalf of scrum team?**
Answer: As the scrum master acts as a coach for his team, he should motivate his team to perform every task. Although he can remove impediments on behalf of scrum team but he should not do this. It is recommended for a scrum master not to over pamper nor overrule the team. There may be something when the team can face failure, at that time the scrum master should help them. He should guide them with an appropriate method t0 get out of the problem. Scrum master should prompt his team members to become independent enough to face problems and take a decision by themselves. This is one of the frequently asked scrum master interview questions, so prepare now and get ready to answer.

**35. What is the difference between the agile & traditional way of working?**
Answer: This question is to judge whether one is aware of the environment of the agile way of working. Here the answer is expected to cover few or all of below:
The traditional way is sequential where design->Development->Testing etc. happens one after another whereas in agile all of this is done in every iteration/sprint
Changes are welcomed in agile as Scope is flexible whereas in traditional manner scope is fixed in the beginning due to which changes have to follow change request path
Progress is measured with % completion traditionally whereas working software is the measure of progress in agile
Project Manager as a central controlling authority is traditionally driving the project whereas Self-motivated and self-organizing teams drive the projects in agile

**36. What is the difference between Sprint Planning Meeting and Sprint Retrospective Meeting?**
Answer: The difference between Sprint Planning Meeting and Sprint Retrospective Meeting is as follows:
Sprint Planning Meeting – A meeting in which all the Scrum roles (product owner, scrum team, and scrum master) have a discussion about the team's priority features and product backlog items is known as sprint planning meeting. This meeting is held every week and lasts for almost 1 hour.
Sprint Retrospective Meeting – A meeting in which all the Scrum roles (product owner, scrum team, and scrum master) have a discussion about the good part of the sprint, the bad part of the sprint, and the sprint improvements is known as sprint retrospective meeting. This meeting that is held at the sprint review meeting or at the end of the sprint; it lasts for 2-3 hours.

This is one of the frequently asked Agile Scrum interview questions. You may be asked to define the above terms separately or the difference between these two.

### 37. How is an agile testing methodology different from other testing methodologies?

Answer: Agile scrum interview questions may include a number of questions from agile testing. Let's understand how you can answer such questions.
The agile testing methodology involves the division of the whole testing process into multiple small segments of codes. In every step, these segments of codes undergo testing. There are a number of additional processes involved in agile testing methodologies such as team communication, strategic modifications for optimal results and many others.

### 38. What is the difference between agile & scrum?

Answer: This question will test whether you are aware of a broader picture of agile and its different frameworks or flavors. Below image will help you explain this answer where you can say agile is an umbrella of values and principles and under it, one of the lightweight frameworks is called Scrum.

### 39. Do you have a Scrum Master certification?

This question may seem awkward to you but it is one of the most popular Agile Scrum interview questions. If an interviewer asks this question, it doesn't mean that a certification is must for the job position. Just be confident while answering whether you have a scrum master certification or not. If you are a certified scrum master, just share the details of your certification like certification exam, score obtained, and the year of passing the certification exam. In case you don't have a certification, mention and highlight your experience in the particular field. Also, let the interviewer know if you are planning to invest in the certification in the near future.

### 40. Do you hold any agile certification? Why did you choose this certification?

Answer: Agile and scrum certifications are hot in the market and organizations are expecting the candidates to hold one or more out of it. Certifications generally looked by organizations are:

- ACP (Agile Certified Practitioner)
- ASM (Agile Scrum Master)
- CSM (Certified Scrum Master)
- PSM (Professional Scrum Master)
- Safe Agilist

### 41. When can you say your story is ready to develop/groom enough to deliver?

Answer: Ready is a stable state of Scrum that is linked to a user story. As per the Definition of Ready (DoR), a user story have to satisfy some conditions before picking it up for a sprint i.e. to be in the ready state. So, the conditions that are essential for the development/grooming of a user story specify if the user story is ready to develop/groom enough to deliver or not.
Basically, the following questions should be answered to consider a user story ready:
**Why**: Is it clear what the business or stakeholders are trying to achieve?
**What**: Is the goal or outcome of the user story clear?
**How**: Is the strategy for the implementation of user story clear? Is the story is small enough?
The conditions for the user story are defined by scrum master in coordination with the product owner. Although the conditions vary for the different projects, some of the common conditions for user story are –

- It is clear and well-written in a format to identify user type, function, and benefits
- It is self-contained i.e. independent of other user story inherently
- It is small so that can be delivered in a single sprint
- It has a defined acceptance criteria for all the functional requirements and appropriate non-functional requirements
- It should have been estimated by the scrum team
- All the external blocking dependencies should have been resolved before starting the sprint
- The resources/team have all the skills required to deliver the sprint

So, if the user story can give satisfactory answers to the above questions and meet the conditions defined, it is considered to be ready.

### 42. How do you manage if the story is high priority and resources left before last day of sprint completion?

x
Check if you can manage the situation and finish the task by yourself or try to get some other resources to work. (Although this won't be feasible as you will have to first explain everything about the sprint and task done to the new team; of course it won't be possible in a day.) So, if you can't manage to complete the sprint by yourself, it's better to tell this to your product owner. You can ask him to give some more time to complete the sprint, so you can get a new team or get it done by yourself by working extra hours.

### 43. As a scrum project manager, what are your responsibilities?

Answer: Firstly, I would like to correct the question as there is no project manager role in scrum i.e. Scrum Project Manager is not a defined role. The responsibilities of a project manager are split between the scrum master, product owner, and the development team.
Scrum Master is a facilitator who is responsible to manage the development teams working on Agile methodology. He is an intermediate between the product owner and the development team to work for the achievement of the final goal. The scrum master role is similar to the project manager in a few cases, and the responsibilities of a scrum master are:

- Performing Sprint planning

- To schedule the daily Scrum meeting
- Management of responsibilities of the Scrum process
- Assisting Scrum teams to follow Scrum practices
- Work to remove barriers to allow the team focus on work
- Providing assistance with the Product Backlog
- Co-ordinating with Product Owner to design Product Backlog items for the upcoming Sprint
- Motivating team not to be distracted by the external factors
- Helping team to improve the dynamics to reach the goal