1. **What is the waterfall methodology?**

Much like construction and manufacturing workflows, waterfall methodology is a sequential design process. This means that as each of the eight stages (conception, initiation, analysis, design, construction, testing, implementation, and maintenance) are completed, the developers move on to the next step.

As this process is sequential, once a step has been completed, developers can’t go back to a previous step – not without scratching the whole project and starting from the beginning. There’s no room for change or error, so a project outcome and an extensive plan must be set in the beginning and then followed carefully.

**Advantages of waterfall model:**

1>This model is simple and easy to understand and use.

2>It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.

3>In this model phases are processed and completed one at a time. Phases do not overlap.

4>Waterfall model works well for smaller projects where requirements are very well understood.

**Disadvantages of waterfall model:**

1>Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.

2>No working software is produced until late during the life cycle.

3>High amounts of risk and uncertainty.

4>Not a good model for complex and object-oriented projects.

5>Poor model for long and ongoing projects.

6>Not suitable for the projects where requirements are at a moderate to high risk of changing.

**2) When to use the waterfall model:**

This model is used only when the requirements are very well known, clear and fixed.

Product definition is stable.

Technology is understood.

There are no ambiguous requirements

Ample resources with required expertise are available freely

The project is short.

**3) What is Agile model ?**

Agile development model is also a type of Incremental model. Software is developed in incremental, rapid cycles. This results in small incremental releases with each release building on previous functionality. Each release is thoroughly tested to ensure software quality is maintained. It is used for time critical applications.

Developers start off with a simplistic project design, and then begin to work on small modules. The work on these modules is done in weekly or monthly sprints, and at the end of each sprint, project priorities are evaluated and tests are run. These sprints allow for bugs to be discovered, and customer feedback to be incorporated into the design before the next sprint is run.

**Advantages of the Agile Methodology**

1> The Agile methodology allows for changes to be made after the initial planning. Re-writes to the the program, as the client decides to make changes, are expected.

2>Because the Agile methodology allows you to make changes, it’s easier to add features that will keep you up to date with the latest developments in your industry.

3>At the end of each sprint, project priorities are evaluated. This allows clients to add their feedback so that they ultimately get the product they desire.

4>The testing at the end of each sprint ensures that the bugs are caught and taken care of in the development cycle. They won’t be found at the end.

5> Because the products are tested so thoroughly with Agile, the product could be launched at the end of any cycle. As a result, it’s more likely to reach its launch date.

**Disadvantages of Agile Methodology**

1. With a less successful project manager, the project can become a series of code sprints. If this happens, the project is likely to come in late and over budget.

2. As the initial project doesn’t have a definitive plan, the final product can be grossly different than what was initially intended.

**4) When should you use Agile methodology?**

1. When rapid production is more important than the quality of the product.

2. When clients will be able to change the scope of the project.

3. When there isn’t a clear picture of what the final product should look like.

4. When you have skilled developers who are adaptable and able to think independently.

5. When the product is intended for an industry with rapidly changing standards.

**5) What are the roles in Scrum?**

Scrum only prescribes 3 roles - the Product Owner, Scrum Master and the Delivery team.

**6) What is Sprint in scrum ?**

The Sprint is the heart of Scrum. It is a short, consistent cycle no longer than four weeks. The goal is to have an iteration short enough to keep the team focused but long enough to deliver a meaningful increment of work.

1. **What is story point in Agile ?**

Story point is a arbitrary measure used by Scrum teams. This is used to measure the effort required to implement a story. In simple terms its a number that tells the team how hard the story is. Hard could be related to complexity, Unknowns and effort.

**8) What is velocity in agile ?**

Velocity is a measure of the amount of work a Team can tackle during a single Sprint and is the key metric in Scrum. Velocity is calculated at the end of the Sprint by totaling the Points for all fully completed User Stories.

Velocity is the average number of points from that past 3-4 sprints and is used to help predict when backlog items will be delivered.

It also facilitates very accurate forecasting of how many stories a Team can do in a Sprint. (In Scrum this is called using Yesterday’s Weather.) For forecasting purposes the average of the last three Sprint's Velocity should be used. Of course, this means it takes three Sprints of experience for a Team to determine its Velocity accurately, which can sometimes be difficult to explain to impatient stakeholders.

Without Velocity, Release Planning is impossible. By knowing Velocity, a Product Owner can figure out how many Sprints it will take the Team to achieve a desired level of functionality that can then be shipped. Depending on the length of the Sprint, the Product owner can fix a date for the release.

For most agile development teams velocity will typically stabilize between 3 and 6 iterations.

**9) What is user story in Agile?**

a story is represented as an issue, and individual tasks within the story are represented as sub-tasks.

1. **Responsibilities of Scrum Master**

1. Facilitates Scrum Ceremonies like Daily stand up, Spring review, Spring retro spection

2. Encouraging collaboration between the Scrum team and product owner. He is acting as a middle man between development team and Product Owner

3. Scrum master is like a coach who helps his team to perform better. Scrum Master shield the team from any interference.

4. Help Product owner to arrange product backlog to maximase the value

5. He keeps development team focused and self organizaed to achieve high value product.

6. Removing obstacles that affect the team.

7. Make sure scrum is followed properly.

8. Determines Length of Sprint

9. Ensures that the Scrumboard and the Impediment Log remain updated

10. ScrumMaster should update the release burndown chart at the end of each sprint.

**11) Responsibilities of Product Owner**

1. Creates and MAINTAINS the Product Backlog

2. Prioritizes and sequences the Backlog according to business value

3. Participates in the daily Scrums, Sprint Planning Meetings and Sprint Reviews and Retrospectives.

4. Clearly communicate the business requirements to the Team

5. Get detail level of requirement from stakeholders or customers. Build and maintain a relationship with the Stakeholders and explain them about progress of Product at the end of every Sprint.

6. Terminates a Sprint if it is determined that a drastic change in direction is required

7. His job is to define epics/user stories to Agile development team.

8. He is responsible for determining the release date and contents.

**12) What is burn down chart ?**

Progress on a Scrum project can be tracked by means of a release burndown chart. The ScrumMaster should update the release burndown chart at the end of each sprint.

The horizontal axis of the sprint burndown chart shows the sprints; the vertical axis shows the amount of work remaining at the start of each sprint. Work remaining can be shown in whatever unit the team prefers -- story points, ideal days, team days and so on.

A burn down chart displays the amount of work a team has burned through such as hours during the sprint.

**13) Explain Scrum ceremonies**

**1> Sprint planning**

Attendees

Required: development team, scrum master, product owner

When: At the beginning of a sprint.

Duration: Usually an hour per week of iteration–e.g. a two-week sprint kicks off with a two-hour planning meeting.

Purpose: The product owner will have a prioritized product backlog. They discuss each item with the development team, and the group collectively estimates the effort involved. The development team will then make a sprint forecast outlining how much work the team can complete from the product backlog. That body of work then becomes the sprint backlog.

**2>Daily stand-up**

Attendees

Required: development team, scrum master

Optional: team stakeholders, product owner

When: Once per day, typically in the morning.

Duration: No more than 15 minutes. Don't book a conference room and conduct the stand up sitting down. Standing up helps keep the meeting short!

Purpose: Stand-up is designed to quickly inform everyone of what's going on across the team. It's not a detailed status meeting. Have each team member answer the following questions:

What did I complete yesterday?

What will I work on today?

Am I blocked by anything?

There's an implicit accountability in reporting what work you completed yesterday in front of your peers. No one wants to be the team member who is constantly doing the same thing and not making progress.

**3>Sprint review**

Attendees

Required: development team, scrum master, product owner

Optional: project stakeholders

When: At the end of a sprint.

Duration: 30-60 minutes.

Purpose: Sprint review is a time to showcase the work of the team. They can be in a casual format like "demo Fridays", or in a more formal meeting structure. This is the time for the team to celebrate their accomplishments, demonstrate work finished within the iteration, and get immediate feedback from project stakeholders. Remember, work should be fully demonstrable and meet the team's quality bar to be considered complete and ready to showcase in the review.

**4>Retrospective**

Attendees

Required: development team, scrum master, product owner

When: At the end of an iteration.

Duration: 60 minutes.

Purpose: Agile is about getting rapid feedback to make the product and development culture better. Retrospectives help the team understand what worked well–and what didn't.

Retrospectives aren't just a time for complaints without action. Use retrospectives to find out what's working so the team can continue to focus on those areas. Also, find out what's not working and use the time to find creative solutions and develop an action plan. Continuous improvement is what sustains and drives development within an agile team, and retrospectives are a key part of that.

**14) Differance between product backlog and sprint backlog**

The product backlog is a priority list of user requirements, use cases to be done in order to create, maintain and sustain a product. Product Owner owns the product backlog,(s)he is the one who prioritize it based on the customers feedback or business value. It is a very active document where all the wishlist and user requirements are gathered.

Sprint backlog is the subset of the product backlog. Each sprint, scrum team picks the user stories from product backlog on top of its stack, the number of user story picked by scrum team for a time box sprint is based on the average velocity of a scrum team. Sprint backlog is an output of a sprint planning meeting.