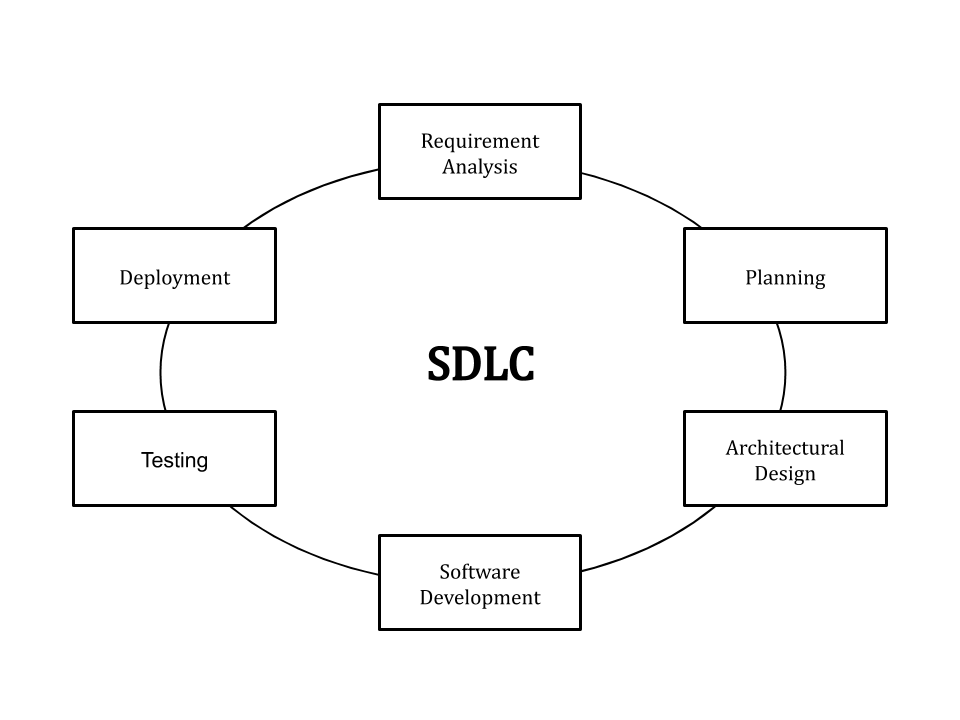
1. Explain SDLC at a high level

Software Development Life cycle (SDLC) is the process of developing the software and delivering it through different stages. SDLC stages covers the complete life cycle of a software. The purpose of the SDLC is to deliver high-quality product as per the requirement of the Customer.

The different phase of SDLC are

1. Requirement Analysis/gathering
2. Planning
3. Design
4. Development
5. Testing
6. Deployment

The SDLC model includes Waterfall model, Spiral model and Agile model.



**Requirement Analysis:**

During this phase, all the relevant information is collected from the customer to develop a product as per their expectation. Any ambiguities must be resolved in this phase only.

Business analyst and Project Manager set up a meeting with the customer to gather all the information like what the customer wants to build, who will be the end-user, what is the purpose of the product. Before building a product a core understanding or knowledge of the product is very important.

**Planning:**

In this stage of the SDLC, the team determines the cost and resources required for implementing the analysed requirements. It also details the risks involved and provides sub-plans for softening those risks.

In other words, the team should determine the feasibility of the project and how they can implement the project successfully with the lowest risk in mind.

**Design:**

 This phase of the SDLC starts by turning the software specifications into a design plan called the Design Specification. All stakeholders then review this plan and offer feedback and suggestions. It’s crucial to have a plan for collecting and incorporating stakeholder input into this document. Failure at this stage will almost certainly result in cost overruns at best and the total collapse of the project at worst.

**Develpoment:**

In this phase coding starts once the developer gets the Design document. The Software design is translated into source code. All the components of the software are implemented in this phase.

**Testing:**

Testing starts once the coding is complete and the modules are released for testing. In this phase, the developed software is tested thoroughly and any defects found are assigned to developers to get them fixed.

Retesting, regression testing is done until the point at which the software is as per the customer’s expectation. Testers refer SRS document to make sure that the software is as per the customer’s standard.

**Deployment:**

Once the product is tested, it is deployed in the production environment or first [UAT (User Acceptance testing)](https://www.softwaretestinghelp.com/what-is-user-acceptance-testing-uat/) is done depending on the customer expectation.

In the case of UAT, a replica of the production environment is created and the customer along with the developers does the testing. If the customer finds the application as expected, then sign off is provided by the customer to go live.

1. **What is waterfall and why it is still relevant?**

The Waterfall model is a classical mode used in software development life cycle with a linear and sequential approach .The Waterfall model is the earliest SDLC approach that was used for software development.

Waterfall model develops software systematically from one phase to another in downward fashion. This model is divided into different phase and the output one phase is used as input for next phase. Therefore every phase has to be completed before next phase starts.



The different phases of Waterfall model as follows

1) **Requirement:**

The potential requirements of the application are methodically analysed and written down in a specification document that serves as the basis for all future development. The result is typically a requirement document that defines what the application should do, but not how it should do it.

**2) Design:**

The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

**3) Development:**

With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.

**4) Testing:**

With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.

**5) Deployment:**

Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.

**6) Maintenance:**

 There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

Waterfall model is still relevant as it can still provide benefits, particularly for larger projects and organizations that require stringent stages and deadlines.  Waterfall method does allow the project as a whole to maintain a more detailed, robust scope and design structure due to all the upfront planning and documentation stages. This is particularly well suited to large teams that may see members come and go throughout the life cycle of the project, allowing the burden of design to be placed on the core documentation and less on any individual team member.

**4) Who is Scrum Master?**

The scrum master is the person on the team who is responsible for managing the process, and only the process. They are not involved in the decision-making, but act as a lodestar to guide the team through the scrum process with their experience and expertise.

Not everyone on the team will have the same understanding of scrum, and that’s especially true for teams new to the framework. Without a scrum master promoting and supporting the process, who can help team members understand the [theory, practice, rules and values of scrum](https://www.projectmanager.com/blog/scrum-methodology), the project can flounder and fail.

The scrum master has several roles in the project. The scrum master serves the product owner by making sure that the goals, scope and product domain are clear to everyone on the scrum team. They offer techniques and [tools to manage the product backlog](https://www.projectmanager.com/software) effectively and help the scrum team know that there is a need to keep the product backlog items clear and concise.

When it comes to the scrum team, a scrum master acts as a coach, helping them to self-organize and [work cross-functionally](https://www.projectmanager.com/blog/6-tips-developing-cross-functional-teams). They also assist with getting the team to create a high-value product by removing obstacles in their process and coaching them through meetings or other venues when help is needed.

The scrum master also helps the organization by leading and coaching the transition into a scrum framework. This includes helping the employees and [stakeholders](https://www.projectmanager.com/blog/what-is-a-stakeholder) work in an empirical product development mode. In this capacity, the scrum master will lead change that increases the productivity of the team while working with other scrum masters to help foster scrum throughout the organization.

Scrum master performs some or all of the following

1. **Stand-ups:** Performs stand ups as required.
2. **Sprint planning meetings:** Decide the estimation of the project.
3. **Sprint reviews:** Take part in meetings and receive feedback.
4. **Retrospectives**:Areas of improvement are taken into consideration and actions are planned for future sprint.
5. **Board Administration**: Acts as administrator of the scrum board.
6. **One on one:** Arranging one on one’s with team members and stakeholders individually as required. This is important for team development and getting to know each other.
7. **Blockers:** Scrum Master helps team to in removing external and internal blockers by process and workflow improvement.
8. **Reporting:** Give an account of the ongoing work.

**5.** Differentiate between Product/Sprint Backlog

Product Backlog is essentially a list of all prioritised 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.

The items on the Product Backlog are often estimated by the development team(s) in Story Points or Ideal Time, which is a size estimate, relative to the other items on the list. As User Stories and Features get closer to development, they are typically refined and broken down into smaller User Stories or tasks.

Product managers and Product owners should care about all details that breakdown each item into steps, making all tasks easier for developers to understand. It’s also important to pay attention to time estimates to help teams determine how soon to start every project.

* It is a very active document where all the wish list and user requirements are gathered
* Product owner makes sure that content of product backlog “user stories” are defined in detailed level
* user story in product backlog should be enough in sizing to be fit in one sprint
* All aspects like use case scenarios, condition of satisfaction aka acceptance criteria should be provided in each of the user stories
* The product backlog acts as an input to the sprint backlog when comes to functionality
* There are also bugs/issues, epic, user stories and themes are included in the product backlog

Sprint Backlog is subset of the product backlog. Each sprint, scrum team picks user stories from product backlog on top of its stack, the number of user stories picked by scrum team for a time box sprint is based on average velocity of a scrum team. Product owner set the sprints goal for the team, scrum team pick the user stories from product backlog fulfilling these goals.

* Sprint backlog is dynamic in nature, each sprint the above scenario is repeated. Good practice is to keep the sprint backlog aka sprint goal as static as possible during a sprint.
* During each sprint planning session, the team returns back to product backlog to pick recently prioritized user stories for the sprint.
* In Sprint backlog, scrum team works on how the user stories would be implemented in a sprint by dividing it further into tasks and estimating it.

6) What is Epic & Story?

An epic is large user story that cannot be delivered within single iteration. The epic is that large which can split into small stories.

Epic allow you to keep track of large, loosely defined ideas in backlog without the need to over populate your backlog with multiple items.

Epics allow you a way to establish a hierarchy for your backlog items where the Epic represents the original idea often closely related to a particular outcome. The user stories associated with that epic represent the various aspects of the solution you need to deliver, or the options you have for satisfying that need.

A story is a functional requirement that provides some business value. It also has to be small enough to comfortably fit within a sprint.

Stories are written in a language that is easily understood by the Product Owner and business users. That way they can understand the progress that has been achieved by completing the story.

7) What is called Velocity in SCRUM ?

Velocity is the measure of work completed by the development team within each sprint, which a repeated cycle typically spanning two to four weeks. In Scrum, project work is broken down into userstories, which focus on specific functionality for an end user. The development team estimates the time and effort needed to develop and test each user story with points, or a numeric value. The work completed is the summation of these assigned points for user stories that have been fully developed and tested.

8) Explain the SCRUM ceremonies

SCRUM ceremonies play vital role in agile software delivery process. These SCRUM ceremonies facilitate frameworks for team to get work done in structural manner give assistance to set expectations, empower the team to collaborate effectively and ultimately gives results.

The four Scrum ceremonies are

a) **Sprint Planning**: Sprint planning is the scrum ceremony outlined to make sure the team is prepared to get the right things done every sprint. This Scrum meeting happens at the beginning of a new sprint and designed for Product Owner and Development Team to meet and review prioritized Product Backlog. Through series of discussion and negotiations, the team should ultimately create a sprint backlog that contains all items they are committing to complete at the end of sprint.

b) **Daily Scrum meeting:** The Daily Scrum is team’s chance to get together, define a plan for the day’s work and identify any blockers. This Scrum ceremony provides frequent opportunity for the team to get together and communicate individual progress toward the sprint goal. The Scrum Master is responsible for clearly roadblocks for Development Team so they can focus on delivering the work identified in sprint planning.

c) **Sprint Review**: The Sprint Review is Scrum ceremony where all work completed during the sprint can be displayed to stakeholders. This led the stakeholders to see things sooner than later and inspect the product. All the work showcased during this time should be fully demonstrable and meet the definition. For Sprint Review the scrum team includes product owner, development team and scrum master.

d)**Sprint Retrospective meeting**: This is the final scrum ceremony that allows the team to review the work that’s just completed and identify the areas of improvement. It provides platform to the Scrum team to discuss what’s going well and suggestions for changes.

9. What is grooming?

Grooming is the meeting of Scrum team in which the product backlog items are discussed and the next sprint planning is prepared. Product grooming is critical in product management because it means keeping the backlog up-to-date and getting backlog items ready for upcoming sprints.

The grooming includes splitting big items into smaller ones, rewriting backlogs to be more expressive, deleting obsolete or more needs items and so on.

Product owners identify user stories based on priorities for the next sprint planning. Groomed [Scrum backlog](https://hygger.io/product-backlog/) helps to streamline sprint planning meetings to avoid stretching them for hours.

Permanent grooming meetings save time for the development team for further discussion during sprint cycle because they give a clarity to developers and testers about the requirements.

The grooming process gives product owners, managers or business analysts more chances to enhance the requirements with more information if it’s required.

10. How Jira board is effective in SCRUM?

Scrum is an agile methodology where products are built in a series of fixed-length iterations. There are four pillars that bring structure to this framework: sprint planning, stand ups, sprints, and retrospectives. Out-of-the-box, Jira Software comes with a comprehensive set of agile tools that help your scrum team perform these events with ease.

Jira makes backlog the centre of your sprint planning meeting, so you can estimate stories, adjust sprint scope, and check velocity and reprioritize issue in real-time.

Jira has several tools that can help sprint planning by following

Version management, Easy backlog grooming, sprint planning and story points.

Also daily scrum, stand-ups are short mini-meeting where the team gathers to go through a quick list of what's been done, what they will work on next and where they require someone's input or help. Jira Software helps your team get a quick snapshot of the work in progress, so you can come prepared to discuss the most critical items for the day.

Jira Software has a number of agile reports specific for scrum teams. For example, Burn down and Velocity charts, give your team critical insight into their agile process. Reports make retrospectives more data-driven and highlight areas to improve for upcoming sprints.

11. Differentiate between SCRUM & Waterfall

**Waterfall is the traditional approach to software development. It is based on the principles of sequence, strict hierarchy, and documentation.**

* Each Waterfall project includes at least five sequential stages: the project’s planning, designing, performing, testing, and deploying. Usually, there are even more sequential stages in the Waterfall projects. None of them can begin before the previous stage is finished.
* It is also impossible to return to the previous stage in the Waterfall methodology. It means that if a fail occurred at the early stages of the Waterfall process and was identified only at the final stage (testing), the whole project should be made from scratch even after all the work was finished.
* Constant use of documentation is one more specific feature of the Waterfall method. The documents are required because the customer is not involved in the process of software development. The Waterfall developers should gather all the necessary information about the customer’s wishes before the work on a project starts and document it. All their further work is based on documents.

**The Scrum methodology is based on different principles. It is flexible and does not require strict structure.**

* The Scrum teams are independent. They have no project managers or other formal leaders. The only person who coordinates the work of a Scrum team is a Scrum Master, but he is there as a coach rather than a leader.
* All the work of a Scrum team is subdivided into small charts that are called sprints.
* The intermediate product is shown to the Product Owner after each sprint so he can estimate it.
* The project’s priorities may change after every sprint.

12. Explain the responsibilities of Product Owner

The Product Owner is a member of the Agile Team responsible for defining Stories and prioritizing the Team Backlog to streamline the execution of program priorities while maintaining the conceptual and technical integrity of the Features or components for the team.

The [Scrum](https://www.mountaingoatsoftware.com/agile/scrum) product owner is typically a project's key stakeholder. Part of the product owner responsibilities is to have a vision of what he or she wishes to build, and convey that vision to the scrum team. This is key to successfully starting any agile software development project. The agile product owner does this in part through the product backlog, which is a prioritized features list for the product.

The Product owner has significant role in maximizing the value produced by the team and ensuring stories meet the users need and comply with definition done. For most enterprises moving to Agile, this is a new and critical role.

This role has significant roles and responsibilities outside local team, including working with Product Management, Customers, Business Owners and stakeholders.

Business savvy is important for the agile product owner because he or she is the decision maker regarding what features the product will have. That means, the agile PO should understand the market, the customer and the business in order to make sound decisions.