1) What IS SDLC? (Software development life cycle)

- It is step by step process used by the software industry, within a software organization to design, develop and test high quality software as per requirements or Expectations given by customers within time & cost limits.
- It is one type of structure used for software development in which planning, implementation, testing, documentation, and ongoing maintenance and support.

- ***SIX PHASES OF SDLC PROCESS***

1) **Requirement Collection / Gathering**
2) **Analysis (development)**
3) **Design**
4) **Implementation (development)**
5) **Testing**
6) **Maintenance (development)**

2) What is agile methodology?

- It is combination of iterative and incremental process method, in which we have to focus on process flexibility and customer satisfaction by rapid delivery of working software product.
- Breaks the product in to small incremental builds.
- These builds are provide in iterations.
- Each iteration need almost one to three weeks.
- Each iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing.
- At the end of the iteration a working product is displayed to the customer.
- In agile the tasks divided to time boxes (small time frames) to deliver specific features for a release.
- Each build is incremental and delivered after each iteration.
- In this method process started early and due to its flexibility and adaptability it becomes popular.

Agile manifesto principles

1) **Individuals and interactions** - in agile development, self – organization and motivation are important, as are interactions like co-location and pair programming.
2) **Working software** - demo working software is considered the best means of communication with the customer to understand their requirement, instead of just depending on documentation.
3) **Customer collaboration** - As the requirement cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirements.
4) **Responding to change** - agile development is focused on quick responses to change and continuous development.

Advantages

- It is a very realistic approach to software development.
- Promotes teamwork and cross training.
- Functionality can be developed rapidly and demonstrated.
- Resource requirements are minimum.
- Suitable for fixed or changing requirements.
- Delivers early partial working solutions.

- Good model for environments that change steadily.
- Minimal rules, documentation easily employed.
- Enables concurrent development and delivery within an overall planned context.
- Little or no planning required.
- Easy to manage.
- Gives flexibility to developers.

Disadvantages

- Not suitable for handling complex dependencies.
- More risk of sustainability, maintainability and extensibility.
- An overall plan, an agile leader pm practice is a must without which it will not work.
- Strict delivery management dictates the scope, functionality to be delivered, and adjustment to meet the deadlines.
- Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.
- There is very high individual dependency, since there is minimum documentation qenerated.

- Transfer of technology to new team members may be quite challenging due to lack of documentation.

3) What is SRS?

Software Requirements Specification

- It is the complete description of system behaviour to be developed.
- It includes a set of use cases that describe all of the interactions that the users will have with the software.
- The SRS contains both function and non-functional (or supplementary) requirements.
- Use cases are also known as functional requirements.
- Non-function requirements are requirements which impose constraints on the design or implementation(such as performance requirements quality standards, or design constraints).
- The specification of software requirements are described by IEEE 830-1998.
- This standard describes possible structures, desirable contents, and qualities of a software requirements specification.

Types of Requirements

- Customer Requirements
- Functional Requirements
- Non-functional Requirements

4) What is OOPS?

Object oriented programming systems: *(Black Box Testing)*

5) Basic concept of OOPS

Object

Class

Encapsulation

Inheritance

Polymorphism

Abstract

6) What is Object?

Instances of a class: (An object is a member of a given class with specified vales rather than variables)

: To create memory for the class.

: To access whole the properties of a class except private

7) What is class?

It is a collection of **data member (**variable**)** and **member function (**process, methods**)** with its behaviour.

8) What is encapsulation?

Data hiding (wrapping up of data into single unit)

Private your data member and member function

9) What is Inheritance?

Properties of parent class derived into child class

Properties of super class extends into sub class

: ***Main purpose is : Reusability, Extensibility***

: There are five types of

1) Single
2) Multilevel
3) Hierarchical
4) Multiple : java does not support
5) Hybrid : java does not support

10) What is polymorphism?

Ability to take one name having different forms
Many forms or multiple forms

There are two types
1) Compile time (method overloading)(static Binding)
2) Run time (method overriding) (Dynamic Binding)

11) What is RDBMS?

RDBMS (Relational Bate **B**ase Management System): the software used to store, manage, query, and retrieve data stored in a relational database is called a relational database management system (RDBMS). The RDBMS provides an interface between users and applications and database.

12) What is SQL?

it is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS)

13) Write SQL Commands

1) DDL – data definition language.
   : Create table, create database, use ,truncate etc…
2) DML - data manipulation language.
   : Insert, update, delete.
3) DQL - data Query lang.
   : select
4) DCL/TCL - Data/Transaction control lang.
   : Commit, rollback, grant, etc…

14) Write SDLC Phases with basic introduction

1) Requirement Collection

- It comes from customers side in any format.
- It may be Incomplete, incorrect and unclear.
- It may be change frequently or inadequately captured.
- It needs validation during throughout life cycle process, not only at the time of software release.
- Early prototyping can help clarify the requirements.
- Functional requirements are like a basic needs and stated by user, which can see directly in the final product.(Ex – includes taxes in billing applications)
- Non – functional requirements defined as quality attributes of software. It helps to verify the performance of the software i.e. performance, stress, usability, security testing.
- (Ex – the site should be load in three seconds when the simultaneous users are >10,000.)
- THREE TYEPS OF PROBLEMS can be arise in requirement collections

A) Lack of Clarity – documents is precise and easy to read is very hard to create.
B) Requirement confusion – functional and non functional requirements tends to be intertwined.
C) Requirement Amalgamation – several different requirements may be expressed together.

2) Analysis (Research)

- Analysis phase defined how will be these system requirements will be accomplished independently.
- Customer's problems in the form of requirements will be solved in this phase.
- It is the phase in which to prepare documents clear and precise format.

- The deliverable result at the end of this phase is "Software requirement Specification"
- It is Project Strategy Documents in which we can decide what is to be built.

- In this phase team is decides the Details like Programming Language, Environment, machine, packages, application architecture, Distribute the architecture laying, memory size, platforms and algorithms, data structure, interface and many others Technical details.
- Fine grain analysis of each step can boost efficiency and minimize the cost and efforts.

3) Design

- Prepare the technical details in form of architecture document to guide the developers.
- Clients also final the product design in this phase.
- Developers also defined the technical details of product depending on project.
- Technical details like Screen design, database, sketches, system interface and prototype.
- The architecture team also converts the typical scenarios into a test plan.

4) Implementation

- The Developers team should build exactly what has been requested in design phase & the S.R.S from analysis phase.
- The developers' team builds the components either from sketches or by composition.
- The core development process starts and work is divided in modules/units and actual coding is done.
- Coding is one of the longest phases of SDLC.
- It deals with issues of Quality, Performance, Baseline, Libraries and Debugging.
- Critical error Removal

5) Testing

- Quality is very important
- These is merit in this approach, because it is hard to see one's own mistake and a fresh eye can find obvious errors much faster than the person who has read & re-read the material many times.
- It is a separate phase which is performed by a different testing team after implementation phase.
- In Testing phase testers execute the test cases against the application, report the defects and retested the fixed defects
- Unfortunately, if we apply delegating testing to another team this leads to a slack of attitude to the implementation team toward quality.

- If the team are known as crafts man, then the team should be responsible for establishing high quality across all phases.
- An attitude must be changed to take place guaranteed quality it continuous testing is done
- Regression Testing
- Internal Testing
- Unit Testing
- Application Testing
- Stress Testing

6) Maintenance

- Maintenance is the process of changing system after it has been deployed.
- It is the process of modifying the software product after delivered to the customer or end user.
- To modify and update the software product to correct the bugs and improve the performance.
- It contains configuration and version update, reengineering and redesigning.
- **Once when the customers starts using the developed system then the actual problems comes up and needs to be solved from time to time is known as maintenance.**

- **Corrective maintenance** is modification and updation, by identifying and repairing defects.
- **Adaptive maintenance** is modification and updation, when customers need the product to run on new platforms, on new operation system or run the product on new hardware or software.
- **Perfective maintenance** is implementing new requirements and need to change different functionalities according to the customers demand.
- In a spiral lifecycle, everything after the delivery and deployment of the first prototype can be considered as Maintenance.

15) Draw Use case on online book shopping

- Open browser computer /application in mobile phone
- Login in to website / web application
- Search book of your choice
- Select quantity
- Tap on buy now button
- Enter/select delivery address
- Select type of address (home/office)
- Enter contact details
- Select payment method - COD
- Review order summery and placed order
- Check order id confirmation and arrival date

16) Draw Use case on online bill payment system (paytm)

- Open browser computer / application in mobile phone
- Login in to patym by scan the code / using login id and password or OTP.
- Scroll down and go to recharge & bill payment menu

- Select any bill of service.
- Select details of service provide (like as state, name)
- Enter your service id and view billing amount.
- Select pay now
- Select payment method and account details
- Done the payment and get online receipt same.
- Logout from web browser / go to application home page.

17) Explain phase with of the water full model

- Step by step process between various phase . & we can go only forward in this model

requirement collection
Analysis
Design
Implementation
Testing & maintenance

18) Write phases of spiral model

Planning
Risk analysis
Engineering
Customer evaluation

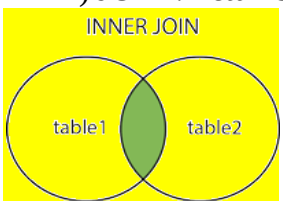
19) What is join?

A JOIN Clause is used to combine rows from two more table, based on a related column between them.

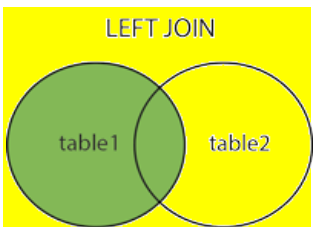
20) Write types of joins

Four different types of the joins in SQL

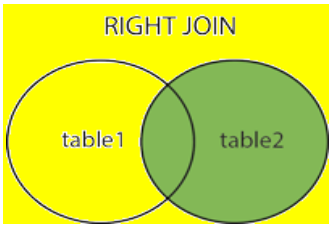
1. (INNER) JOIN : returns records that have matching values in both tables



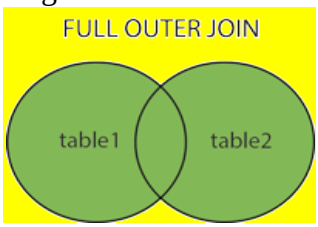
2. LEFT(OUTER) JOIN : Returns all record from the left table, and the matched records from the right table



3. RIGHT(OUTER) JOIN : Returns all record from the right table, and the matched records the left table



4. FULL (OUTER) JOIN : Returns all record when there is a match in either left or right table



21) <u>Draw use case on online shopping product using COD.</u>

- Open browser computer / application in mobile phone
- Login into website / web application
- Search product
- Select quantity
- Select size / colour
- Tap on buy now button
- Enter / select delivery address
- Select type of address (home/office)
- Enter contact details
- Select payment method - COD
- Review order summery and placed order
- Check order id confirmation and arrival date.

22) <u>Draw use can on online shopping product using payment gateway</u>

- Open browser computer / application in mobile phone
- Login in to website / web application
- Search product
- Select quantity
- Select size / colour
- Tap on buy now button
- Enter / select delivery address
- Select type of address (home/office)
- Enter contact detail
- Select payment method by selection of payment gateway (like phone pay, g-pay, pay UPI. etc)
- Enter payment details
- Make payment successful
- Review order summery and placed order
- Check order id confirmation and arrival date.