What is SDLC?

The Software Development life cycle (SDLC) is a well-structured process that guides software development projects from start to finish. It provides a clear framework for planning, building, and maintain software, ensuring that development is systematic and meets quality standards.

Phases of SDLC

1.Requirement Analysis

2.Feasibility study

3.Design

4.Coding

5.Testing

6.Deployment

7.Maintenance

Phase1: Requirement Analysis

→ It is the first phase of SDLC in which all the necessary information is collected from the customer to develop the software as per their expectation.

→Some important question like: what is the need of software, who will be the end-user, what is the future scope of that software etc. are discussed.

→The main aim of this phase is to collect the details of each requirement of the customer so that the developers will clearly understand what they are developing and how to fulfill the customer’s requirements.

Phase2: Feasibility Study

→It is the second phase of SDLC in which an organization discusses about the cost and benefits of the software.

→it is an important phase because profits from the software plays an important role as if cost is very high then company may face loss.

→After the feasibility study, the project may be accepted, accepted with modification or rejected.

→It measures how much beneficial the product is for the organization.

Phase3: Design

→It is the third phase in which architects start working on logical designing of the software.

→In this phase a SRS (System Requirement Specification) document is created which contains all logical details like how the software will look like, which language will be used, database design, modular design etc.

→This phase provides a prototype of the final product.

→Basically, all it includes is design of everything which has to be coded.

Phase4: Coding

→When the designing of the software is completed, then, a group of developers start coding of the design using a programming language.

→The interface of the software and all its internal working according to design phase is implemented in coding phase.

→A number of developers cods the modules and then all modules are arranged together to work efficiently.

→It is the longest phase of SDLC.

Phase5: Testing

→Once the software development is completed, then it is sent to the testers. The testing team starts testing the functionality of the entire system.

→In this phase, the software is checked for bugs or errors.

→Whenever a bug is found, then the software is resent to the coders to fix it and then overall software is re-tested.

→This is done to verify that the entire application works according to the customer requirement.

Phase6: Deployment

→After overall testing of the software and after checking that is bug free, then the software is launched and available for the users to use it.

→Even after deployment of the software, if any bugs or errors are still found then the software is re-evaluated by the maintenance team and then it is re-deployed with a new version.

Phase7: Maintenance

→The maintenance team look over the software usage and users feedback.

→Maintenance is necessary to eliminate errors in the system during its working life and to tune the software.

→The bug fixing upgrade and enhancement of the software is looked over by the maintenance team.

2.What is software testing?

Software testing is the process of evaluating and verifying that a software application or system meets its requirements and functions as expected.it involves testing to identify bugs or defects and ensure quality.

4 levels of Software testing

1.Unit Testing

2.Integration Testing

3.System Testing

4.User Acceptance Testing

1.Unit Testing

▪A unit test is a type of software test that focuses on testing individual components of a software product.

2.Integration Testing

▪Integration testing is the process of checking how well different software modules work together to ensure they interact correctly.

3.System Testing

▪System testing is the process of testing the entire software system to make sure it meets all requirement and works correctly in real world scenarios.

4.User Testing

▪User Acceptance testing is the final stage of software testing where real users check if the software works as intended and meets their business needs before it goes live.

3.What is RDBMS?

A Relational Database Management System (RDBMS) is a software application that manages and organizes data in a relational database. It stores data in tables with rows and columns, allowing for structured access and manipulation through languages like SQL.

4.What is SQL

SQL (Structured Query Language ) is programming language used to interact with database.

5. Write a query to create the table in Structured Query Language.

CREATE TABLE table \_name

(

Column\_name1 datatype constraint,

column\_name2 datatype constraint,

column\_name3 datatype constraint,

);

6.Write a query to insert data into table.

INSERT INTO TABLE\_NAME

(column1, column2,column3,….columnN)

VALUES

(value1,value2,value3,….valueN);

7.Write a query to update data into table with validations.

UPDATE TABLE\_NAME

SET “column\_name1”='value1',”column\_name2”='value2'

WHERE “ID”=”value”

8.Write a query to delete data from table with validations.

DELETE FROM table \_ name WHERE condition;

9.Write a query to insert new column in existing table.

ALTER TABLE table \_ name

ADD COLUMN column \_ name

10.Write a query to drop table and database.

DROP TABLE table \_name;

11.Write a query to find max and min value from table.

SELECT MAX (price) from products;

SELECT MIN (price) from products;

12. what is integration testing?

Integration testing focuses on verifying the interactions between different modules or components of a system. It ensures that individual units work together as expected.

13. What is oops

Oops, in the context of computer programming , stands for object-oriented programming system. It’s a programming paradigm that revolves around the concept of objects , which contain both data (attributes) and methods (functions) that operate on that data. Essentially , it’s a way to structure code by modeling real- world entities as objects and their interactions.

14.what is object

A basic unit of oop ,representing a real world entity with attributes (data) and methods(behavior).

15. What is class

Collection of method, object, datatypes and constructor and variables.

16.what is encapsulation

Bundling data and methods within a class and restricting direct access to data from outside the class ,promoting data security and integrity.

17.what is inheritance

Child class can use the functionality of parent class

18.what is polymorphism

The ability of objects to take on multiple forms, allowing methods to behave differently based on the object they are called upon ,promoting flexibility and code reuse.

19. What is Alpha testing?

Alpha testing is a type of software testing performed to identify bugs before releasing the product to real users or to the public.

Alpha testing is one of the user acceptance testings.

20. what is beta testing?

Performed by actual users in a real-world environment, usually before the final release.

21.what is 7 key principles? Explain in detail?

1.testing shows the presence of defects

Testing can reveal defects, but it cannot prove a software is completely defect-free.

2.Exhaustive testing is impossible

Due to the vast number of possible input combinations and conditions , its not feasible to test every scenario.

3. Early testing

Starting testing early in the development lifecycle helps identify and fix defects sooner, reducing costs and effort.

4.Defect clustering

A small subset of modules or functionalities often contains the majority of defects.

5. pesticide paradox

Repeating the same tests can become less effective at finding new defects over time, requiring test cases to be regularly reviewed and updated.

6. Testing is context dependent

The testing approach should be tailored to the specific project ,system , and context.

7. Absence of errors fallacy

Finding no defects doesn’t necessarily mean the software is usable or meets user needs.

22.Difference between smoke and sanity?

Smoke testing is conducted early in the software build process to ensure that the core functionalities are working and the build is stable enough for further testing.

Sanity testing is performed later to verify that specific changes have not led to new issues ,ensuring the build is stable after modifications.

23.What is Api testing

▪ Api stands for the application programming interface

▪ Basically a collection of functions an procedures which allows us to communicate two applications or libraries.

24. Types of Api Testing

1. simple object access protocol- no protocol follow, low band width call ,less secure

Ex. Json, html, text ,xhtml

2. representational state transfer- protocol follow, high band width call, high secure

Ex.xml

25. Explain phases of the waterfall model

▪The waterfall model was the first SDLC model to be introduced.

▪It is based on the basic concept of SDLC.

▪In waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phase.

▪ The waterfall model illustrates the software development process in a linear sequential flow.it means that every phase in the development process begins only if the previous phase is completed.

▪It is also known as linear-sequential life cycle model.

When to use waterfall model

▪Requirements are very clear and fixed.

▪no ambiguous requirements

▪Tools and technology used is consistent and is not changing

▪Resources are well prepared and are available to use.

▪A project is short

▪Risk is minimum

Advantages of waterfall model

▪simple and easy to understand and use

▪works well for smaller projects where requirements are well understood and fixed

▪ Easy to manage due to the rigidity of the model

▪ phases are processed and completed one at a time

Disadvantages of waterfall model

▪Not the best choice for complex and large projects

▪not suitable for the projects where requirements are at a moderate to high risk of changing

▪Adjusting scope during the life cycle can end a project

26.write phases of spiral model

▪spiral model is a risk handling software development model.

▪It is a combination of iterative nature of prototyping with the controlled and systematic aspects of waterfall model.

▪There are some phase that are completed in one iteration and the output is a small prototype of the large software. Then the same iteration is repeated until the entire software is built.

When to use spiral model

▪This model is best used for large projects which involve continuous enhancements.

▪ it is used where the requirements are of developing a full-fledged, large, complex system with a lot of features and facilities from scratch.

▪ it may used when experimenting on technology, trying out new skills, or where the user is not able to offer requirements clearly.

Advantages of spiral model

▪Bulky and complex system can be made easily because of the risk management factor.

▪ changing requirements can be accommodated.

Disadvantages of spiral model

▪Management is more complex

▪End of the project may not be known early

▪Not suitable for small or low risk projects and could be expensive for small projects

27.Explain working methodology of agile model and also write pros and cons.

▪Mostly used model in todays digital era.

▪Agile means “ the ability to respond to changes from requirements, technology &people”

▪it is an incremental and iterative process of software development.

When to use the agile model?

1.When project size is large.

2.when frequent changes are required.

3.when a highly qualified and experienced team is available.

4.when a customer is ready to have a meeting with a software team all the time.

5. projects with flexible timelines and budget

Advantages of Agile model

▪support customer involvement and customer satisfaction

▪strong communication of the software team with the customer.

▪little planning required

▪anytime changes are acceptable.

▪provides a very realistic approach to software development

▪it reduces total development time.

Disadvantages of Agile model

▪Due to the lack of project documentation, once the project completes and the developers allotted to another project ,maintenance of the finished project can become a difficulty.

▪Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.

28.What is Exploratory testing?

In exploratory testing, testers explore the application without predefined test cases. This helps find bugs that may be missed during planned testing.

29. What is adhoc testing?

Adhoc testing is unplanned and informal testing performed without any documentation. The tester uses their experience and intuition to find defects.

3o.Difference between verification and validation

Verification

1.It comes before validation

2.It is known as static testing.

3. It is executed by quality assurance team or developers.

4.It includes checking documents, design, code and program.

5. it does not involve executing the code

6.it finds bugs early in the development cycle

Validation

1.it comes after verification

2.it is known as Dynamic testing

3.it is executed by the testing team.

4.it includes testing and validating the actual product

5.it always involves executing the code

6. it can find bugs that the verification process can not catch.

31. difference between priority and severity

Priority

1.priority is define as in which order defect should be fixed

2. priority is set according to how much bug impacting the business

3.priority of defect decided in consultation with the manager/client

Severity

1.severity is define as how strongly bug imapact the software functionality, performance & usability

2.QA engineer determines the severity level of the defect

3.According to severity bug can be critical ,high.

32.What is SRS

▪SRS stands for “system Requirement specification”.

▪ A SRS is a detailed description of a software system to be developed with its functional and non-functional requirements.

▪SRS document contains all logical details like how the software will look like, which language will be used, database design , modular designs etc.

▪SRS is a formal report which acts as a representation of the software.

▪a good SRS defines the how software system will interact with all internal modules, hardware, communication with other programs etc.

▪SRS allows the customers to review whether all their requirement are fulfilled by the software or not.

▪SRS serves two main purposes:

1.it is used to defines the need and expectation of the users.

2.it serves as a contract document between customers and company.

33.What is error, defect, bug and failure?

Errors: are human mistakes made during the software development lifecycle. These can be in the code, design, or even in the requirements gathering process.

Defect: are the consequences of those errors, meaning they are the observable problems or deviations from the expected behavior of the software.

Bug: a bug is a flaw or fault in a computer program or system that causes it to produce an incorrect or unexpected result , or behave in unintended ways.

Failure: is when the software doesn’t perform its intended function due to an error or defect.

34.What is white box testing and list the types of white box testing ?

▪It is a software testing technique in which internal structure, design and coding of software are tested.

▪it is done to verify the flow of input -output in the software and to improve design, usability and security.

▪in this testing , programming skills are required to design test cases.

▪it focuses on checking of predefined inputs against expected and desired outputs.

35. What is black box testing?

▪it is a software testing technique in which the functionalities of software are tested without having knowledge of internal code structure, implementation details and internal paths.

▪it is entirely based on software requirements and specifications and mainly focuses on input and output of software.

36. What is GUI testing?

GUI testing (Graphical User Interface Testing) is a process of testing an application’s user interface to ensure it meets specifications and works as expected. The main goal of GUI testing is to verify that the application’s visual elements, such as buttons, menus, icons, text fields, function correctly and provide a seamless user experience.

37. What is join?

▪Join means to combine something.

▪A join clause is used to combine data from two or more tables , based on a related column between them

TYPES OF JOINS

1.INNER JOIN

2.LEFT JOIN

3. RIGHT JOIN

4.FULL JOIN

INNER JOIN: Returns records that have matching values in both tables

LEFT JOIN: Returns all records from the left table, and the matched records from the right table

RIGHT JOIN: Returns all record from the right table, and the matched records from the left table

FULL JOIN: Returns all records when there is a match in either left or right table

38.What is load testing?

▪The load testing is the most important essential part of performance testing.

▪Load testing is used to check the performance of an application by applying load like less than or equal to the desired load.

▪the load is a quantity, which means it only focuses on the number of users.

39. What is stress testing?

▪the stress testing is testing ,which checks the behavior of an application by applying load greater than desired load.

▪since it is non- functional testing, so we use this testing when the application is functionally stable.

40.Write a scenario of Door

1.verify the door opens when a valid access card is presented.

2. verify the door remains locked when an invalid access card is used.

3. verify the door closes automatically after a set time.

4. verify the door stays open when the “hold open” function is activated.

5.verify the door enters “fail-secure” mode during a power failure.

6.verify the door unlocks during a fire alarm or emergency signal.

7. verify the system logs each access attempt.

8. verify the door cannot be opened manually when in locked state.

9. verify the door stays locked during system maintenance mode.

10.verify an alert is generated if the door is forced open.

40.write a scenario of ATM

1.verify the ATM accepts a valid ATM card and prompts for pin.

2. verify the ATM rejects an invalid or unreadable card.

3. verify the ATM accepts a valid pin and allows access to account functions.

4.verify the ATM rejects an incorrect pin after 3 attempts and blocks the card.

5. verify that the user can withdraw cash within the available account balance.

6. verify that withdrawal fails if the requested amount exceeds the account balance.

7. verify that the ATM does not dispense cash if the machine runs out of cash.

8. verify that a transaction receipt is printed after a successful transaction.

41. write a scenario of pen

1. verify that the pen writes smoothly on standard paper.

2. verify that the ink flows consistently without skipping.

3.verify that the pen starts writing immediately without shaking.

4. verify that the pen does not leak link under normal storage conditions.

5.verify that the link dries within a reasonable time to avoid smudging.

6. verify that the pen’s cap fits securely and does not fall off easily.

42. write a scenario of microwave owen

1.verify that the microwave starts when a valid time and power level are set.

2. verify that the microwave does not start when the door is open.

3.verify that the microwave stops automatically when the timer reaches zero.

4. verify that pressing the “stop” or “cancel” button stops the operation immediately.

5.verify that the microwave light turns on during operation and off when idle.

6. verify that the turntable rotates when the microwave is running.

7. verify that the door locks securely when closed and unlocks smoothly when opened.

8. verify that the microwave beeps when the timer ends.

43. write a scenario of coffee vending machine

1. verify the machine displays the welcome screen and menu options when powered on.

2. verify that the user can successfully select a drink from the menu.

3. verify that the machine accepts valid currency/coin/note/card and updates the balance correctly.

4. verify that the machine rejects invalid currency or insufficient payment.

5. verify that the selected drink is dispensed after payment is completed.

6. verify that the machine stops dispensing if the cup is not placed correctly.