Q1 What is SDLC

- Software development life cycle

- SDLC is structure imposed on the development of software product that defines the process for planning, implementation and support, and there are number of different development model

- Software development life cycle is series step or phases that provide for development

-there some phases in below

Requirement

analysis

design

development/implementation/code

testing

maintenance

Q2 What is software testing

--> software testing is part of to used identify the correctness, completeness, and quality of development computer software

--> software testing is part of software development process

-->software testing activity to defect and identify the defect in software

--> to release quality product to the client

Q3 write the SDLC phases with basic introduction

* SDLC phases

requirement

analysis

design

development/implementation/code

testing

maintenance

(1) requirement

- establish customer need

- usage scenarios

- features

- requirement will change !

- plan for change

- function requirement

- non-function requirement

- early prototyping [e.g.,]can clarify the requirement

(2) Analysis

-model and specify the requirement -" what"

-this document states in clear what is built

-represent what phases

-the deliverable design document is the architecture

- details on programming language and environment, machines, packages, memory size, platform, interface, data structures, global many other are establish

(3) Design phase

-design Architecture document

-implementation plan

-critical priority analysis

-performance analysis

-test plan

- the architecture team also typical scenarios a test plan

(4) implementation phase

-implementation =code

-critical error removal

(5) testing phases

- simply quality is very important

- customer satisfied with quality of product will remain loyal and wait for the new functionality in the next version

- unit testing

-stress testing

- application testing

- internal testing

- regression testing

-testing phase is performed by a different team after the implementation is completed

(6) Maintenance phase

-updating all analysis , design and user documentation

- corrective maintenance: identifying and repairing defects

- adaptive maintenance: adapting the existing solution to the new platforms

- perfective maintenance: implementing the new requirement in spiral lifecycle, everything after delivery and deployment can be considered “maintenance”

Q.4 Explain phases of the Waterfall model

* Waterfall model (classical software life cycle)
* the classical software life cycle is a the software development as a step-by-step “ waterfall ” between the various development phases
* this flow like water

requirement

analysis

design

implementation

testing

maintenance

* requirement must be frozen to early in the life cycle
* requirement are validated too late
* Applications (when to use )
* product definition is stable
* requirement are very well and documented, clear and fixed
* the project is short
* technology is understood and not dynamic
* Advantages of waterfall model
* quality of the product is good
* since requirement changes are not allowed so finding buges will less
* process and results are well documented
* phases are processed and completed one at the time
* clearly defined stages
* easy to arrange tasks
* Disadvantages of waterfall model
* high amounts of risk uncertainty
* not good model for complex and objects-oriented project
* requirement changes are not allowed
* total investment is more because time to take for rework on defect is time consuming to high investment
* testing start only after the coding
* it is difficult progress within stages

Q.5 write phases of Spiral model

1. Planning = determination of objectives, alternatives and constraints
2. Risk analysis = analysis of alternatives/resolution pf risk
3. Engineering = development of the “next level” product
4. Customer evaluation = assessment of the results of engineering

* Application
* For medium to high risk project
* Significant change are expected in the product during development cycle
* When costs and there are a budget constraint and risk evaluation is important
* Released in multiple version
* Advantages of spiral model
* Testing in done in every cycle before going to next cycle
* Changing requirement can be accommodated
* Allow for the extensive use to prototypes
* Users see the system early
* Disadvantages of spiral model
* Management is more complex
* Spiral may go indefinitely
* Process is complex
* Rechanges are not allowed in between in cycle
* End of project may not be known early

Q.6 What is Agile methodology

- Agile SDLC model is combination of iterative and incremental process model

-rapid delivery software product

- these builds are provided in iteration

- each iteration take time to 1 to 4 weeks

- it divides the software into small increment builds this build are provided in iteration that means big project divides in small project

- each iteration involves all team member working on areas like planning, analysis, design, coding all working

- after the release we check the feedback of deployed software

- if any enhancement needed in the project then it’s done and it’s re-released

Q 7 Explain working methodology of agile model and also write the pros and cons

- Agile SDLC model is combination of iterative and incremental process model

-rapid delivery software product

- these builds are provided in iteration

- each iteration take time to 1 to 4 weeks

- it divides the software into small increment builds this build are provided in iteration that means big project divides in small project

- each iteration involves all team member working on areas like planning, analysis, design, coding all working

- after the release we check the feedback of deployed software

- if any enhancement needed in the project then it’s done and it’s re-released

> Advantage of agile model

* Frequent delivery
* Face to face to communication with the customer
* Less time
* Adaptability
* Promotes teamwork and cross training
* Suitable for fixed and changing requirement
* Resource requirement are minimum
* Easy to manage
* Gives flexibility to developer
* Disadvantages of agile model
* Less documentation
* Maintenance problems
* Not suitable for complex dependencies
* Transfer of technology to new member may be quite challenging due to lack of documents

Q 8 What is SRS

* Software requirement specification
* SRS complete description of the behavior of the system to be developed
* Use case also known as functional requirement the SRS also contain non functional
* Non functional known as ( performance , quality , or design etc. )
* In includes a set of use case that describe all the interactions that the users will have with software
* This standard describes possible structures and quality of a software requirement specification

Q.9 What is oops

* Object oriented programming
* Programming is like writing
* If you can write demonstration you can make a program
* So, programming is also easy
* Object oriented programming is way of writing the programs in organized way object are like a black box where data are hidden
* Learning and practice is necessary

Q.10 Write the basic concepts of oops

* 1) class
* 2) object
* 3) inheritance
* 4) polymorphism

1. Over ridding
2. Over loading

* 5) Encapsulation
* 6) Abstraction
* This all is basic concepts of oops

Q.11 What is class

* Class is collection of data member and member function
* Blueprint for an object
* Ex= class is making all function and data

Q.12 What is object

* Object give permission to access functionality of class

Q.13 What is Inheritance

* Making class from an existing class. Deriving the attribute of some other class
* Inheritance means that one class inheritance the characteristics of other class. this is also called a “is a relationship”
* Ex grandparent

|

Parent

|

Child

Q.14 What is encapsulation

* The process of wrapping the data in a single unit to secure the data from outside world
* In simple word data hiding from other and make it private

Q.15 What is polymorphism

* One name multiple form

1. Over riding = same name of function with parameter but definition will be different
2. Over loading

* Function over loading =same name different parameter
* Constructor over loading = same constructor name but different parameter
* Operator over loading = using the operator to add objects instead variable operands