Software testing assignment

Module-1(fundamental)

1) What is SDLC?  
-> The software development life cycle is the cost-effective and time-efficient process that development teams use to design and build high quality software. The goal of SDLC is to minimize project risks through forward planning so that software meets customer expectations during production and beyond.

2) What is software testing?  
-> software testing:  
 - Testing is a process rather than a single activity.  
 - Software testing is a part of software development process.  
 - Software testing is an activity to detect and identify the defects in the software.  
 - The objective of testing is to release quality products to the client.

3) What is agile methodology?  
-> The agile methodology is a project management approach that involves breaking the project into phases and identify continuous collaboration and improvement.  
-> Agile thought process had started early in the software development and started becoming popular with time due to its flexibility and workableness.

4) What is SRS?  
-> Software Requirement Specification is complete specification and description of requirements of software that needs to be fulfilled for successful development of software system. These requirements can be functional as well as non-functional depending upon type of requirement. The interaction between different customers and contractor is done because its necessary to fully understand needs of customers.

5) What is oops?  
-> Identifying objects and assigning responsibilities to these objects.  
-> objects communicate to other objects by sending messages.  
-> Messages are received by the methods of an objects.  
-> An object is like a black box.  
-> The internal details are hidden.

6) Write Basic concepts of oops?  
-> OOPS:  
 - Classes/objects   
 - Encapsulation/data hiding  
 - Inheritance   
 - Polymorphism   
 - Interface/Methods

7) What is object?  
-> An object represents an individual, identifiable item, unit, or entity, either real or abstract, with a well-defined role in the problem domain.  
- This is the basic unit of object-oriented programming(oop)  
- That is both data and function that operate on data are bundled as a unit called as object

8) What is class?  
-> When you define a class, you define a blueprint for an object.  
- A class represents an abstraction of the object and abstracts the properties and behaviour of that object.  
- In the case of a car or laptop, there will be a blueprint or design created first and then the actual car or laptop will be built based on that.  
- We do not actually buy these blueprints but the actual objects.

9) What is encapsulation?   
-> Encapsulation is the practice of including in an object everything it needs hidden from other objects. The internal state is usually not accessible by other objects.  
- Encapsulation in java is the process of wrapping up of data (properties) and behaviour (methods) of an object into a single unit; and the unit here is a class (or interface).

10) What is inheritance?   
-> Inheritance in oop = When a class derives from another class. The child class will inherit all the public and protected properties and methods from the parent class. In addition, it can have its own properties and methods. An inherited class is defined by using the extends keyword.

11) What is polymorphism?  
-> poly refers to many. That is single function or an operator functioning in many ways different upon the usage is called polymorphism.  
- E.g., the message display details () of the person class should give different results when send to a student object (e.g., the enrolment number)  
-The ability to change from is known as polymorphism

14) Write SDLC phases with basic introduction   
->

|  |  |
| --- | --- |
| Requirement  Gathering | Establish Customer Requirement |
| Analysis | Model And specify the requirements-  “What” |
| Design | Model And Specify a Solution – ‘’Why’’ |
| Implementation | Construct a solution in software |
| Testing | Validate the solution against the requirements |
| Maintenance | Repair defects and adapt the solution to the new requirement |

15) Explain phases of waterfall model  
->

Requirement Gathering

Maintenance

Deployment

Testing

Coding

Design

Analysis

1.Requirement Gathering – All possible requirements of the system to be the system to be developed are captured in this phase and documented in a requirement specification document.

2.Analysis – The requirement document delivered by the requirement phase and maps the requirement into architecture.

3.Design – The specifications from first phase are studied in this phase and the system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

4.Coding – 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.Testing – All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

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

7.Maitanance – 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.

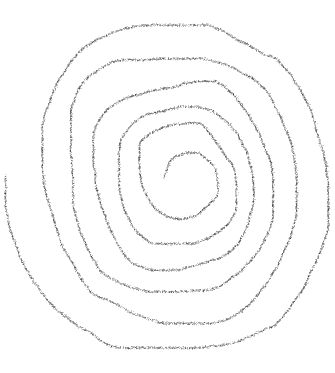
16) Write phases of the spiral model   
->

Risk analysis

Planning



Initial requirements Risk = something that will delay project for increase its cost

Completion go, no-go decision

Alpha demo first prototype

Engineering

Customer Evaluation

1. Planning – determination of objects, alternatives and constraints   
2. Risk Analysis – Analysis of alternatives and Identification/Resolution of risks  
3. Customer Evaluation – Assessment of the results of engineering   
4. Engineering – Development of the ‘’next level’’ product

17) Write agile manifesto principles   
-> Agile Principle -   
1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.  
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.  
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.  
4. Business people and developers must work together daily throughout the project.  
5. Build a project around motivated individual. Give them the environment and support they need, and trust them to get the job done.  
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.  
7. Working software is the primary measure of progress.  
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.  
9. Continuous attention to technical excellence and good design enhances agility.  
10. Simplicity- the art of maximizing the amount of work not done-is essential.  
11. The best architectures, requirements, and designs emerge from self-organizing teams.  
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

18) write working methodology of agile model and also write pros and cons  
- > The agile methodology is a project management approach that involves breaking the projects into phases and emphasizes continuous collaboration and improvement.  
Pros -   
1. Is a very realistic approach to software development   
2. Promotes teamwork and cross training.  
3. Functionality can be developed rapidly and demonstrated.  
4. Resource requirements are minimum.  
5. Suitable for fixed or changing requirements.  
6. Delivers early partial working solutions.  
7. Good model for environments that change steadily.  
8. Minimal rules, documentation easily employed.  
9. Little or no planning required.  
10. Easy to manage  
11. Gives flexibility to developers  
Cons -  
1. Not suitable for handling complex dependencies.  
2. More risk of sustainability, maintainability and extensibility.  
3. An overall plan, an agile leader and agile PM practice is must without which it will not work.  
4. Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the directions.  
5. Depends heavily on customer interactions, so if customer is not clear, team can be driven in the wrong direction.  
6. There is very high individual dependency, since there is minimum documentation generated.  
7. Transfer of technology to new team members may be quite challenging due to lack of documentation.

12) Draw use case on online book shopping

Online book store

<<extend>>

Client

<<extend>>

Admin

13) Draw use case on online bill payment system (Paytm)

system

<<Include>> <<include>>

<<Extend>>

<<Extend>> <<Extend>>

<<Include>>

Supplier

Manager

Inventory system

Cashier

Customer

19) Draw use case on online shopping product using COD.

Delete item in cart

Manage Cart

Edit item in cart

Client register

Add item to cart

View Items

Authentication

Make Purchase

**New User**

iIdentity Provider

Customer

Checkout

**Returning user**

Check register

cCredit Payment

Paypal

20) draw use case on online shopping product using payment gateway.  
->

Login

ADD category

ADD item

ADMIN

USER

Manage item

Manage order

Registration

View item

Make order

Make Payment

Change Password