Module 1: Interview Question Answer

-------------------------------------------------------------------------

1. what is manual testing?

->Manual testing is a software testing process in which test cases are executed manually without using any automated tool.

------------------------------------------------------------------------2) what are the advantages and disadvantages of manual testing?

Advantages: - Disadvantages: -

|  |  |
| --- | --- |
| * Lower Costs in the short-   term   * More Focus on complex   problems   * Well Able to identify   Underlying User issues   * More Flexibility * No programming skills   Needed | * Maintaining Manual   Testers are Expensive   * Extremely Time-   Consuming   * Higher Chances of Human   Error   * Not Good For performance   testing   * Tedious & Monotonous |

-------------------------------------------------------------------------

1. Who are manual testers? What is its role and responsibility?  
   -> The manual tester creates test plans for both new and exciting software. It is also the responsibility of the tester to plan for debugging the code.

------------------------------------------------------------------------4) What is software Testing?

-> Software testing: -

- Software testing is a part of software development process.

- Software testing is an activity to detect and identify the defects in the software.

---------------------------------------------------------------------------------------------------------5) What is testing objectives/purpose?

-> The objective of testing is to release quality products to the client.

Quality is like this in testing:

* Bug-free
* Delivered on time
* Within budget
* Meets requirements /or Expectations
* Maintainable

---------------------------------------------------------------------------------------------------------6) What is difference between bugs errors defects and failure?

-> Differences:

Error: -

* Error is a human mistake which is done by the developer.

Bugs(defects): -

* While testing the software we are checking the functionality of our application whether it is working or not on customer requirements if something is not working on requirements which comes under bugs and defects.

Failure: -

* After releasing the product to the customer and customer is using the software or as a user I’m working with the software and in a real environment where I found some bugs, mismatches and something is not working that is come under the failure.

-------------------------------------------------------------------------7) What is static testing?

-> Static testing is a software testing technique which is used to check defects in software application without executing the code. Static testing is done during verification process.

For example: reviewing, walkthrough, inspection, etc.

-------------------------------------------------------------------------8) What is dynamic testing?

-> In dynamic testing the software code is executed to prove the result of running tests. it’s done during validation process.

For example: unit testing, integration testing, system testing, etc.

-------------------------------------------------------------------------9) When to start testing?

-> Testing can be started from the requirements gathering phase and continued till the deployment of the software.

------------------------------------------------------------------------10) When to stop testing?

* All the high priority bugs are fixed.
* The rate at which bugs are found is too small.
* When the project budget runs out.
* When we run out of time.
* When we have reached an acceptable level of risk.
* When all the defects have been found.

------------------------------------------------------------------------11) 7key principal name?

1. Testing shows presence of Defects
2. Exhaustive Testing is impossible!
3. Early testing
4. Defect Clustering
5. The Pesticide Paradox
6. Testing is Context Dependent
7. Absence of Errors Fallacy

-----------------------------------------------------------------------12) What is defect clustering?

-> Bugs are not often distributed evenly throughout an application. Defect clustering simply means that a small number of features have caused the majority of quality issue in an application.

-----------------------------------------------------------------------

13) What is pesticide paradox?

-> If the same tests are repeated over and over again, eventually the same test cases will no longer find new bugs.

-------------------------------------------------------------------------

14) Explain three layers of architecture?

1. Presentation layer:

* It is also known as the client layer.
* The top most layer of an application.
* By using this layer, we can access the web pages.
* The main function of this layer is to communicate with the application layer.
* This layer passes the information which is given by the user in terms of keyboard action, mouse clicks to the application layer...
* In simple words, it is to view the application.

1. Application layer:

* It is also known as business logic layer which is also known as the logical layer.
* This layer acts as a mediator between the presentation and the database layer. Complete business logic will be written in this layer.
* In simple words, it is to perform operations on the application.

1. Database layer:

* The data is stored in this layer.
* The application layer communicates with the database layer to retrieve the data.
* It contains methods that connect the database and performs required action e.g.: insert, update, delete etc.

-------------------------------------------------------------------------

15) Types of software architecture

1. One tier Architecture:

(ex.:MP3 players, MS office)

1. Two tire Architecture:  
   (ex. Physical Withdrawal in the bank)
2. Three tire Architecture:  
   (ex. Online transfer money)
3. N tier Architecture:  
   (Shopping cart web application)

-------------------------------------------------------------------------

16) What is the difference between developer vs tester?

Developer

1.Software developers, as name suggest, is person who is responsible for writing and maintaining source code of computer programming to develop software.

2.developer should have programming skills, proficiency at writing code, time management skills, etc.   
  
3.its mains aim to make software that is free from errors and bugs.

4.they mainly focus on user’s requirement while developing software

Tester

1. Software tester, as name suggests, is person who is responsible for identifying correctness completeness and quality of developed software.

2.Testers should have deep knowledge of system that is being developed, good communication skills, critical thinking, etc.  
  
3.Its aim is to find bugs and errors in software application if present.  
  
4.they mainly focus on behavior of end user while testing software application.

-------------------------------------------------------------------------

17) What is use case testing?  
-> Use case testing is generally a part of black box testing and that helps developers and testers to identify test scenarios that exercise the whole system on each transaction basis from start to finish. Business experts and developer must have a mutual understanding of the requirement, as it’s very difficult to attain.

* Use case testing is a functional testing technique that helps in identifying and testing scenarios on the whole system or doing start-to-end transactions.
* It helps to identify the gaps in software that might not be identify by testing individual components.
* It is used to develop test cases at the system level or acceptance level.

-------------------------------------------------------------------------

18) what is sdlc? Explain it   
-> SDLC (Software development life cycle) is used in Every software development company because it is the root of the development cycle, if that model would not exist in the world, firstly no software can build secondly if any how it would be made, it’s not going to succeed it has no use, because of no maintenance, but Luckily SDLC model exist in Tech world but why we need it actually!

-------------------------------------------------------------------------

19) explain waterfall model and their advantages and disadvantages?  
-> The classical software lifecycle models the software development as a step-by-step  
‘’waterfall” between the various development phases.

* The waterfall is unrealistic for many reasons,

especially:

* Requirements must be ‘’frozen" to early in the life cycle
* Requirements are validated too late

Disadvantages

1.error can be fixed only during the phase

2.it is not desirable for complex project where requirement changes frequently

3.testing period comes quite late in the developmental process

4.documentation occupies a lot of time of developers and testers

Advantages

1.before the next phase of development each phase must be completed

2.suited for smaller projects where requirements are well defined

3.they should perform quality assurance test (verification and validation) before completing each stage

4.elaborate documentation is done at every phase of the software’s development cycle

20) What you mean by functional requirements and non-functional requirements?

-> Functional Requirement:

These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

-> Non-functional Requirement:

These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called no-behavioural requirements. They basically deal with issue like:

* Portability
* Security
* Maintainability
* Reliability
* Scalability
* Performance
* Reusability
* Flexibility

------------------------------------------------------------------------

21) Explain three types of maintenance

* Corrective Maintenance:  
   Corrective maintenance is what you need to do when something breaks; it is better known as repairs. …
* Preventative Maintenance:  
   Preventative maintenance attempts to spread out the costs by planning activities on a regular basis. …
* Predictive Maintenance:

-------------------------------------------------------------------------

22) Explain v model

The v-model is a type of SDLC model where process executes in a sequential manner in V-shape. It is also known as verification and validation model. It is based on the association of a testing phase for each corresponding development stage. Development of each step directly associated with the testing phase. The next phase starts only after completion of the previous phase i.e., for each development activity, there is a testing activity corresponding to it.

The V-Model is a software development life cycle (SDLC) model that provides a systematic and visual representation of the software development process. It is based on the idea of a “V”’ shape, with the two legs of the ‘’v’’ representing the progression of the software development process from requirements gathering and analysis to design, implementation, testing, and maintenance.

-------------------------------------------------------------------------

23) Explain agile model

* Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product.
* Agile Methods break the product into small incremental builds
* These builds are provided in iterations.
* Each iteration typically lasts from about one to three weeks.
* Every 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 iteration, a working product is displayed to the customer and important stakeholders.

-------------------------------------------------------------------------

24) When risk is high in your project at that which model you have to choose? Write a name and explain it

-> The Spiral model is often used for complex and large software development projects, as it allows for a more flexible and adaptable approach to software development. It is also well-suited to project with significant uncertainly or high levels of risk.

Spiral is one of the most important software development life cycle models, which provides support for Risk handling. In its diagrammatic representation, it looks like a spiral with many loops. The exact number of loops of the spiral is unknown and can vary from project to project. Each loop of the spiral is called a phase of the software development process.

1. Planning  
   2. Risk analysis   
   3. Engineering   
   4. Evaluation   
   5. Planning

-------------------------------------------------------------------------

25) Project base and product base application means?   
-> Project:  
 - If software application is developed for specific customer based on their requirements, then it is called project.  
 A. Project based company:  
 ->mostly deal with projects.  
 Ex. Tcs, Wipro, Infosys

-> Product:  
 - If software application is developed for multiple customers based on market requirements, then it called product.  
 B. Product based company:  
 Ex. Goggle, Microsoft etc

-------------------------------------------------------------------------

26) Explain all opp concept class, object, encapsulation, polymorphism and abstraction   
 a) 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

The two parts of an object

Object = Data + Methods

or

*to say the same differently*

An object has the responsibility to know and the responsibility to do.

b) 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.  
- An object is a particular instance of a class which has actual existence and there can be many objects (or instances) for a class.  
- 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.

c) 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).

d) Abstraction:  
 Abstraction is the representation of the essential feature of an object. These are ‘encapsulated’ into an abstract data type.  
- Data abstraction refers to, providing only essential information to the outside word and hiding their background details, I, e., to represent the needed information in program without presenting the details.

e) 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.

-------------------------------------------------------------------------

-------------------------------------------------------------------------