1. What is SDLC ?

* System Devlopment Life Cycle

1. Request Gathering And Analysis

- get client requirement

- requirement should must be possible

- team work should be possible on requirement

1. Design

- framework database use

- architecture of project

- it is called system architecture

- future scope of architecture

1. Development

- code by developer

1. Testing

- find bug

5. Deployment

- which code linked on google by developer

6. Maintance

- changes

- cost of all things

1. 4 pillar of software testing

* Collection of code which we wil use for particuler task/functionality
* Code that work of particuler function

1.unit testing

2.integration testing

3.system testing

4.uat (user acceptance testing)

1.unit testing :-

- single functionallity testing at a time

2.integration testing :-

- one or more functionallity merge test at a time

3.system testing :-

- website testing all functionallity test at least 3 times start to end

4.uat :-

- application test by client

- quality assorance certified

1. Database testing :-

Database - collection of data store in well (proper) strucure from

Developer backend

RDBMS – Relational Database Management System

Database of rd bms :-

Mysql,oracl,postgree,sqlite

SQL- Structure Query Language

- use language in database

- no sql

- json (java script object notaion)

- mongodb

* View : -
* View is a vertual table
* Original table and vertual table both are connected with each other

QUERY : -

* Create database database name
* Drop database database name
* Create table tablename
* Drop table tablename
* Create view viewname
* For perticuler things delete from table

-delete from tablename where price = 4

-select \* from tablename

When \* is all column

-select \* from tablename where pid=6

Perticuler column

-update tablename set pname=”tv”,price=”5000” where pid=3;

* Update and delete only one time at a one query.
* Insert into tablename(empname ,emdesignation,

empaddress)VALUE(“drumil”,”xyz”,”ahmd”);

* Select customer.cname,customer.caddress,

orders.pname,orders.custid from customer

INNER JOIN orders ON customer.cid=orders.custid

4.what is software ? types of software.

-> collection of code that code we will use for perticular task/functionality.

-> code that work of particuler function.

1.system software:-

-> multiple tasks

Ex:- chips,operating system,log hitory

2.application software:-

-> all appsperticular task/functionality.

Ex:- whatsapp,instagram

5.what is the different between blackbox testing ,whitebox testing, and grabox testing?

-> Blackbox testing (BM):-

- No knowladge of internal code.

- as like manual tester

- no code

-> Whitebox testing (WA):-

-inernal code knowladge available and access.

-automation tester

-JS,JAVA,PYTHON

-> Graybox testing (GMA):-

- no fully access only limited normal access

- combination of blackbox and whitebox.

- for loop,variable datatype

6.smoke vs sanity testing.

->smoke:-

-Developer 100 page website/app-play store.

-Converted in 1 single page.

-Before build

-Collection of file its called build

->sanity:-

-after build storage data

* Cookie:-

Temporary file client side data store.

* Gui testing:-

Graphical user interface

1.responsive testing – bootstrap framework.

2.color and font testing

3.hover->outside design when click

4.label testing

7.7 key principal of s/w testing:-

1.testing shows the presence of defects:-

2.exhustive testing is not possible:-

Before test case –login-positive and nagative

Testcase:-all of the functionalities crite area’s positivity and nagativities

Precondition:- preplaning

Positive:-valid

Nagative:-invalid

3.early testing:-

Static testing – documentation testing

Early phases:- first phases

4.defect clustering:- group

Small small functionality create bugs.

5.principal testing:-

Pastisize testcase paradocks update strongly after use function for change mathod or add new test cases.

6.testing is context dependent:-

Testing is depends on context by different devices and database application.

7.absence of error falacy:-

Future customers requirement fullfil only bugs not require clients requirement is also required.

8. database testing:-

Join :- Merge two tables.

-min 2 tables

-1 column both tables common

- Customer-customer details store

- cid,cname,cmobile,caddess.

- Order-order details store

- Orderid,ordername,order date.

-> when main table’s primary key add in next able that it convert in foraign key.

1.inner join

2.left join

3.right join

4.full join

1.inner join:-

Both table’s common data.

2.left join:-

Left side data

3.right join:-

Right side data

4.full join:-

My sql does not supported

9.API :- applicaion programming interface.

api

Web -------->

<-------- server

Advantage:-

1.api is language independent

2.use functionality code not access

\* there are two types of api

1.rest api

2.soap api

\*rest api:-

-representational state transfer

-free rules ragulation by backend developer

-low net speed to open website

-not secure

-Json,xml,html,text

\*soap api:-

-rules ragulation follow complesory

-secure high

-high net seed to open website

-xml

\*link api given by backend developer And data use fruntend developer .

Json:-java script object notation

<id> </id>

Xml -> xstensable markup language

Api method (query):-

Get ->select data put

Post ->insert

Put ->update all columns

Delete ->delete

Patch ->update perticuler columns

* Exploratory testing:-

Test case changes after testing by tester.

* Monkey testing:-

Unknown testing

10.software requirements specification:- SRS

SRS :- made by business administrator

Admin panel:- handle hole website

Hole data writen in srs

* Static testing:-documentation testing font
* Introduction:-
* Purpose
* Use case :- website flow client
* Uml:- unifide modeling language

Define without code to client

* Usecase diagram:- actor roll
* Class diagram :- table of database
* Activity diagram:- actrol roll in details
* Siquence diagram:-flow wise
* Package diagram:- raw material of projet like (bunch of all functionality which are use for project)
* Deployment diagram:-last result website live on server by diagram it’s process called deployment diagram.

\*SSL:-http->https

\*TLS:-tansaction

SRS:- overview of website or project document.

11.localzation testing:-

-> website in multiple lanuage by location like states.

-> current location :-language set by lacation.

-> country wise multiple language by state.

\*globalization testing:-

-> uber,amazon,alibaba.

-> world wise multiple language by country.

-> language,currency.

\*mutation tetsing:-

-> changes in website

\*gorila testing:-

-> click more times at a time to crash website like 500 times

\*adhoc testing:-

->unknown testing like (without project material)

12.static testing dynamic testing:-

\*static testing:-

-> before website create testing

-> documentation without website open tetsing.

\*dynamic tetsing:-

-> after website open testing.

\*review:-

-> test case checked by senior tester than do final.

\*walkthrough:-

-> final test case represent in team.

\*inspection:-

-> critical project for

-> project manager arranged meeting.

-> perticuler things testcase.

-> than discuss with all meeting member.

13.OOPS concept:-

->class,object,encapsulation,polymorphysom,inheritance,abstaction

\*class:-

- Collection of datatype ,variable.

\*object:-

- Object store non primative data types.

- non primative means which data can not easily provide by programing language.

\*encapsulation:-

-data wraping.

Class

Method

Data

\*inheritance:-

Child parent

->child class can use all functionality of parant class.

->using extends keyword

->class child extends parent { }

->code reuseability

\*code optimization:-

->small code ccomparition between all participate.

\* single:- 1 parant 1 child

\*multiple:- 1 child and parent multiple

\*multilevel:- follow junration.

\*hybrid:-combination of any 2 inherintance.

\*heirarchical:-multiple child 1 parent.

\*polymorphism:-

->one interface multiple implements.

->one person different implements.

\*method overloading:-

->method same name same argument different.

\*method overriding:-

->method same name same argument same.

\*abstraction:-

->sensitive data hide.

->combination of abstract class and abstract method.

14. bug status:-

\* new:- as a tester first bug in entery in exel it’s called new.

\* assigned:- assigned by tester to developer.

\*open:- when developer check the status it’s called open status.

\*in progress:- work in progress.

\*fix:- work is done.

\*retest:- retest by tester for more bug is available or not.

\*close:- if no bug found than close.

\*reopen:- if bug is not solved than reopen.

\*reject:- bug assingment by tester but bug not except by developer.

\*deferred:-it’s not a bug it’s a work of functionality.

15.what is manual tesing ?

->testing by human not need of tools.

\*disadvantages:-

->time consuming.

->bug chance high.

->performed by non tech / tech person.

->performance (timimg of process load page) testing is not possible.

\*advantage:-

->not require technical skills.

->not require tools.

->functional testing is possible.

\*perfomence testing:- automation.

1.load testing

2.stress

3.volume

1.load testing:- automation.

Capacity test by tester generate space for users by developer(increse checked).

2.stress:- automation.

Checked by increse and decres.

3.volume:- depends on database.

Performance chacked by space (capacity).

\*retesting:-

Bug assigned to developer and after fix it and test again.

\*regression:-

After retesting process edd new things and test again.

16. QA vs QC:-

QA(quality assurance):-senior

-experence

-process oriented

-testing process created by qa

-review

-test plan documentation.

QC(quality control):-junior

-product oriented

-test case

-bug find

-bug report generate

17.TEST PLAN :-

Suggest by QA

Suggest by QC

\*advantage of automation tesing:-

->performance testing.

->human involment very less.

->process time fast.

->bug chances high.

->experience.

\*disadvantage:-

->expensive

->high skilled

->depreciate

18.BVA:-

\* Boundry value analysis

-critearea like 18.

-uperlimit lawerlimit.

-17 18 19

-18 60

-17 18 59 60 61