

Q) Explain in detail about Defect life cycle in software testing?

A) Defect life cycle (or) Bug life cycle in software testing is the specific set of states that defect (or) bug goes through in its entire life.

→ The purpose of the defect life cycle is to easily coordinate and communicate current status of defect which changes to various assignees and make the defect fixing process systematic and efficient.

→ Defect status (or) bug status in defect life cycle is the present state from which the defect or a bug is currently undergoing.

→ The goal of the defect life cycle is to better track and understand the actual progress of the defect life cycle.

Defect status workflow

→ 1. New → 2. Assigned → 3. Open → 4. Fixed → 5. Pending Retest → 6. Re-test

2. Assigned

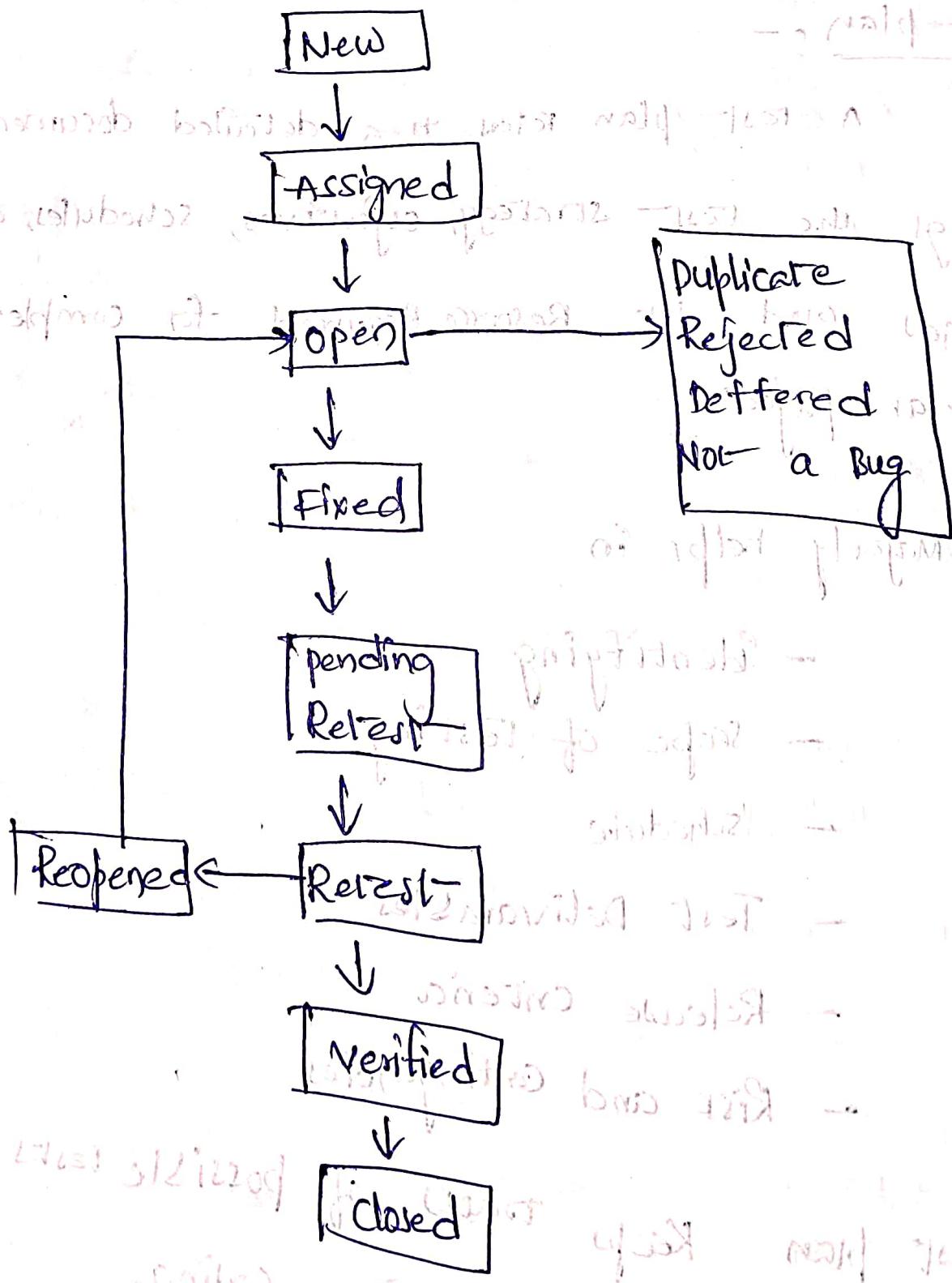
3. Open

4. Fixed

5. Pending Retest

6. Re-test

life cycle



2. Explain the components of test plan. (in detail)

→ Test plan :-

A test plan refers to a detailed document that catalogs the test strategy, objectives, schedules, estimation deadlines and the resource required for completing the particular project.

⇒ It mainly helps in

- Identifying
- Scope of testing
- Schedule
- Test Deliverables
- Release criteria
- Risk and contingencies

⇒ The test plan keeps track of possible tests that

will be run on the system after coding

⇒ Test plan ensures all functional and design

Requirements are implemented as specified in the documentation

⇒ Test plan is a document that develops as the project is being developed.

⇒ It serves a valuable record of what testing has been done.

Components of test plan

1. Test plan identification :-

A unique identifier is to be allocated so that the test plan document can be distinguished from all other documents.

2. Brief introduction :-

A summary of the software to be tested. Definition etc.,

3. Items to be tested :-

A comprehensive list of software items that are to be tested is to be documented in the list of software application areas that is the objectives of the testing.

4. Features to be tested :-

A comprehensive list of characteristics of all the items to be tested. These include functionality, performance, security, portability, usability etc.

5. Features not to be tested :-

Features not to be excluded not to test.

6. Approach of testing :-

It covers the overall approach to testing that ensure that all items and their features will be adequately tested.

7. Acceptance criteria :-

It describes the criteria for determining whether each test item has passed or failed.

8) Suspension criteria and Resumption criteria :-

It describes different conditions under which testing will be suspended and the subsequent condition under which testing will be resumed.

9) Test deliverables :- It describes the documents expected

to be created as a part of the testing process.

10) Testing tasks :- It describes set of tasks required to perform the testing.

11) Environmental Requirements :- It specifies the environment required to perform the testing. The testing include hardware, software, communication facilities, tools, people etc.

12) Responsibilities :- It identifies the individuals or group of people responsible for executing the testing related tasks.

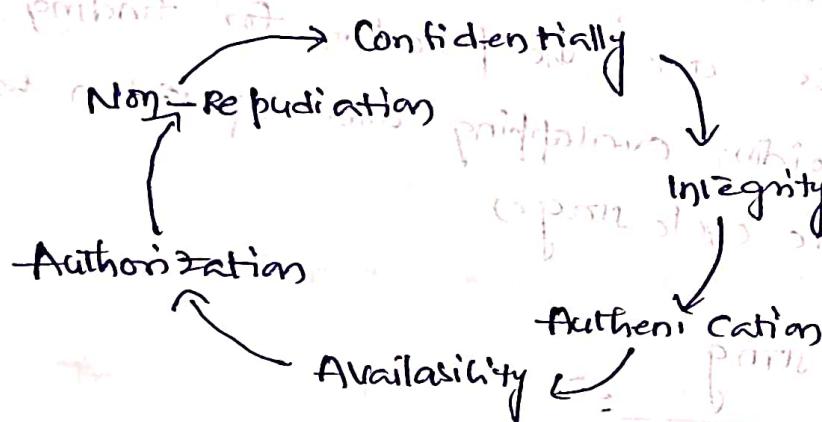
13) manpower and training needs :- It specifies the number and types of persons required to perform the testing, including the skills needed.

14) Schedule of testing :- Defines the important key milestones and dates in the testing process.

15) Risk and Contingencies :- Identifies high-risk assumptions of the testing plan, specifies prevention and mitigation plans for each of them.

- 1) Approval ~~responsibility~~ ~~for~~ ~~the~~ ~~plan~~ ~~is~~ ~~given~~ ~~to~~ ~~the~~ ~~managers~~ ~~and~~ ~~responsible~~ ~~of~~ ~~every~~ ~~individual~~ ~~who~~ ~~must~~ ~~approve~~ ~~the~~ ~~plan~~
- 2) Explain the importance of security testing? write the various areas which has to be focused during security testing?
- 3) Security testing is the process of "testing the authentication of an application to check how well the application is secured from unauthorized users"

principles of security testing :-



areas in security testing :-

1. system software security :-

In this, we will evaluate the vulnerabilities of the application based on different software such as OS, database etc.

2. network security :-

In this, we will check the weakness of the network structure, such as policies and resources.

3. Server side application security :-

We will do the server-side application security

To ensure that the server encryption and its tools are sufficient to protect the software from any disturbance.

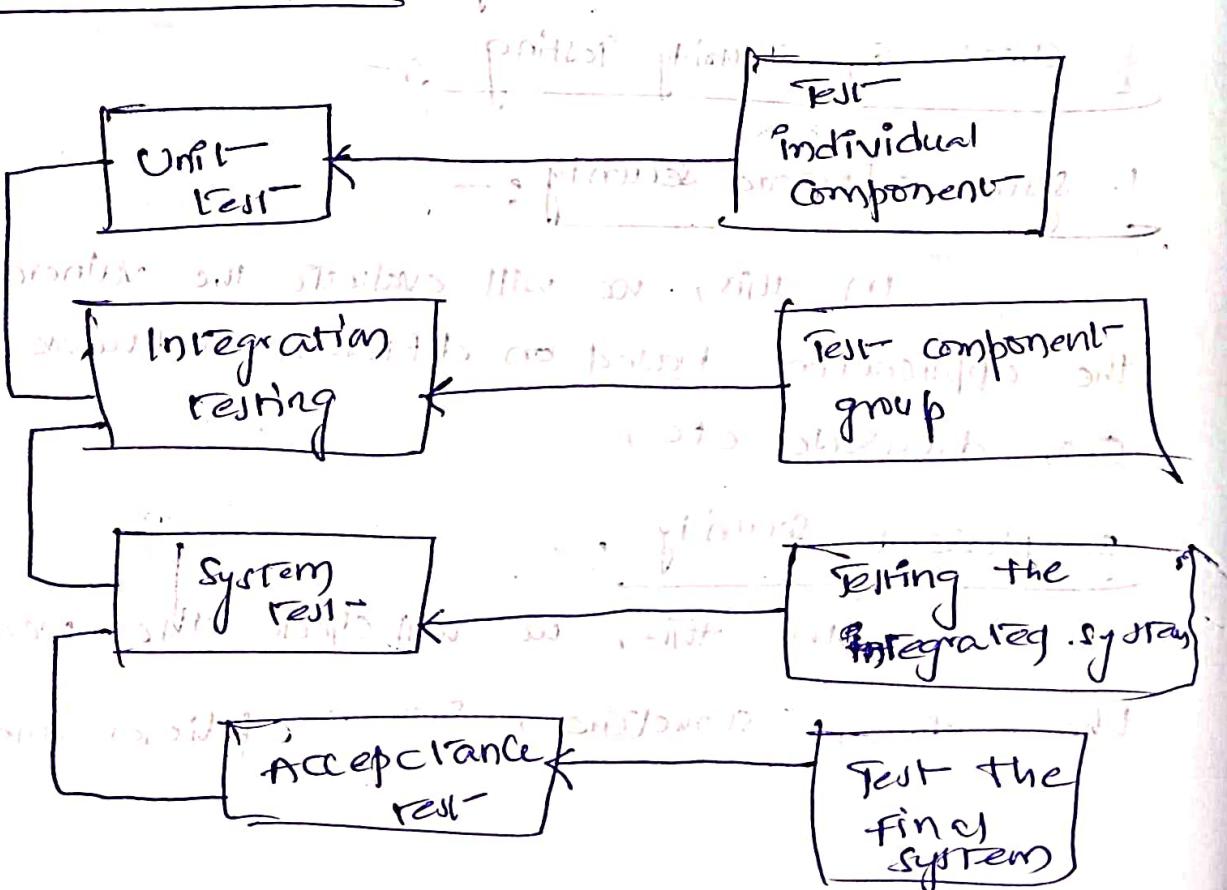
4. Client-side application security :-

In this, we will make sure that any intruders cannot operate or any browser or any tool which is used by customers.

Q5) Explain different levels of testing?

A) Testing levels are the procedure for finding the missing areas and avoiding overlapping and repetition between the development life cycle stages

levels of testing

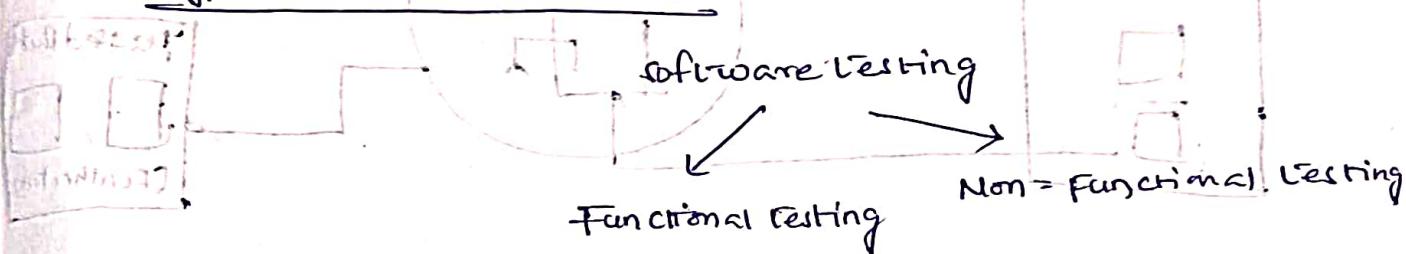


Explain different types of testing?

1. Software testing :- It is the process of evaluating and verifying that a software product / application does what it is supposed to do.

The benefits of software testing include preventing bugs, reducing deployment cost and improving performance.

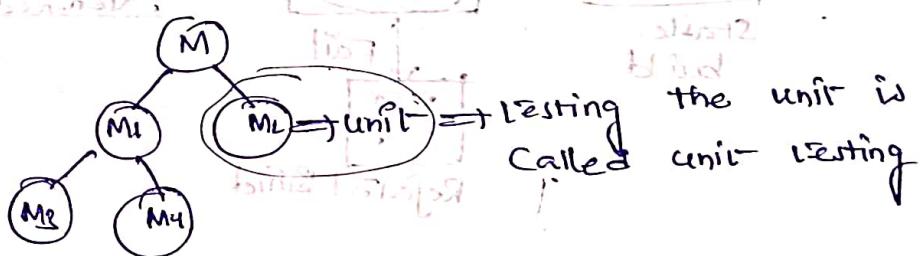
Types of software testing :-



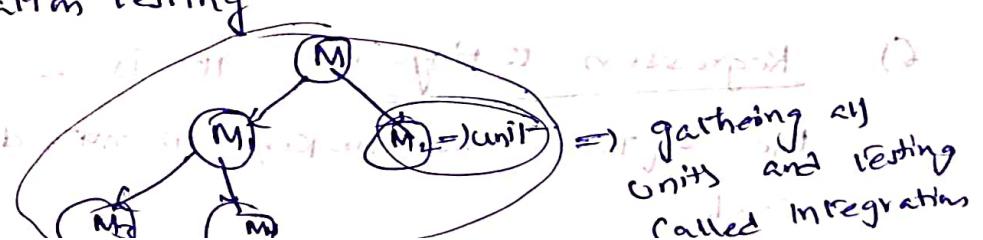
Functional testing, Functional testing is the process that is used to test the features / functions of the software.

Types of functional testing :-

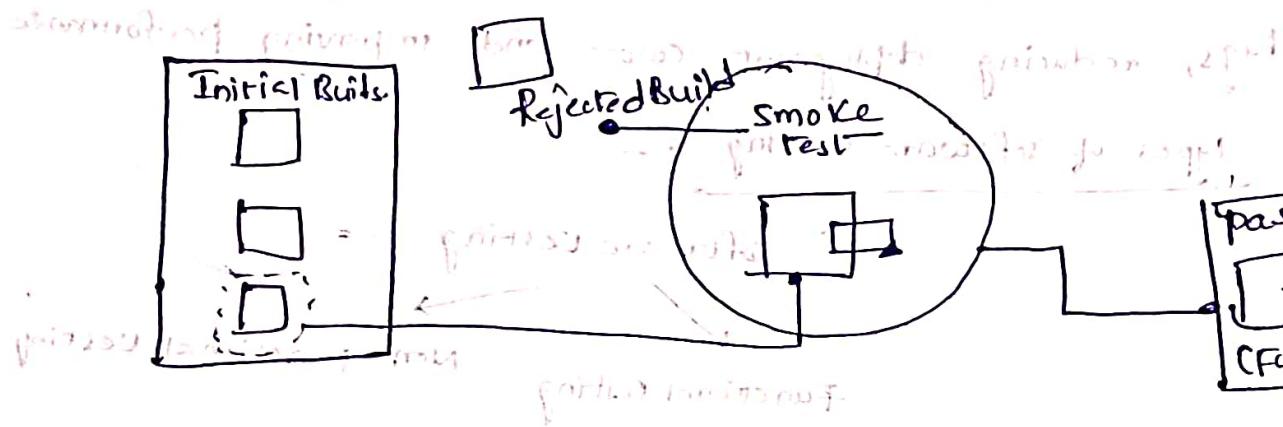
1. unit testing :- unit testing is the process of testing an individual module / unit in a software.



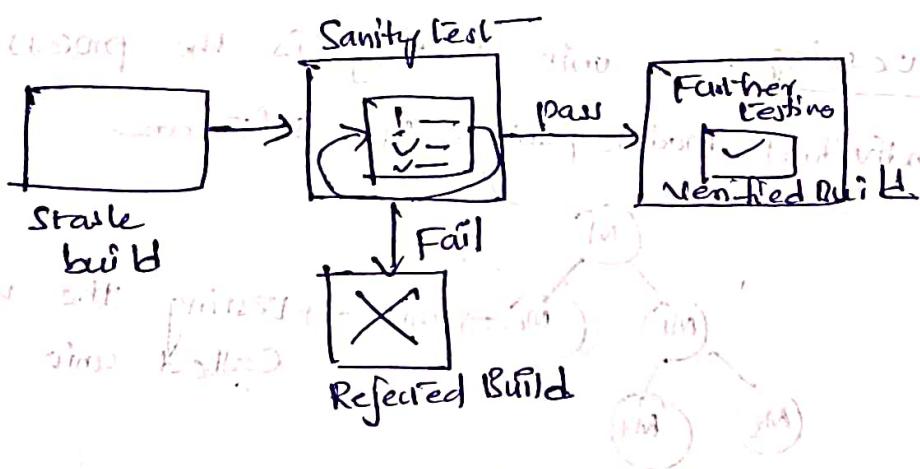
2. Integration testing :- It is the process of combining gathering all unit tests and then testing overall called as integration testing.



3) Smoke testing :- It is also called as "Build verification testing" / Build acceptance testing. It is typically performed at the beginning of the development process to ensure that the most critical functions of an software application are working correctly.



4) Sanity Testing :- It aims to quickly evaluate whether the basic functionality of a new software build is working correctly or not.



5) Re-testing :- Re-testing is the process of executing a previously failed test again on new software to check if the problem is resolved.

6) Regression testing :- It is a process of checking the software is not broken (or not due to changes in code).

- 1) Exploratory Testing :- Testing without any documents (Case 1/1) and identify the functionality of an application by exploring the application.
- 2) Monkey Testing :- The process of testing system under test randomly. The input data that is used to test also generated randomly.
- 3) End-to-End Testing :- Testing the application whether the flow from start to end is same as designed.
- 4) AD Hoc Testing :- The testing is done without a predefined test plan, relying on experience, intuition, and creativity to identify defects and issues.

Non-functional testing :-

- Non-functional testing is a software testing technique that checks the non-functional attributes of the system.
- Types :-
1. Security Testing :- It is basically to check whether the application / product is secured or not and how.
 2. GUI Testing :- used to check the application design of an application.
- Eg:- A hacker can hack the account without any security and authorization.
- Eg:- focus, Countbox, Scrollbar, Move etc.,

3. Usability testing :- It is the process of making an application/app in convenient way to user for usage.

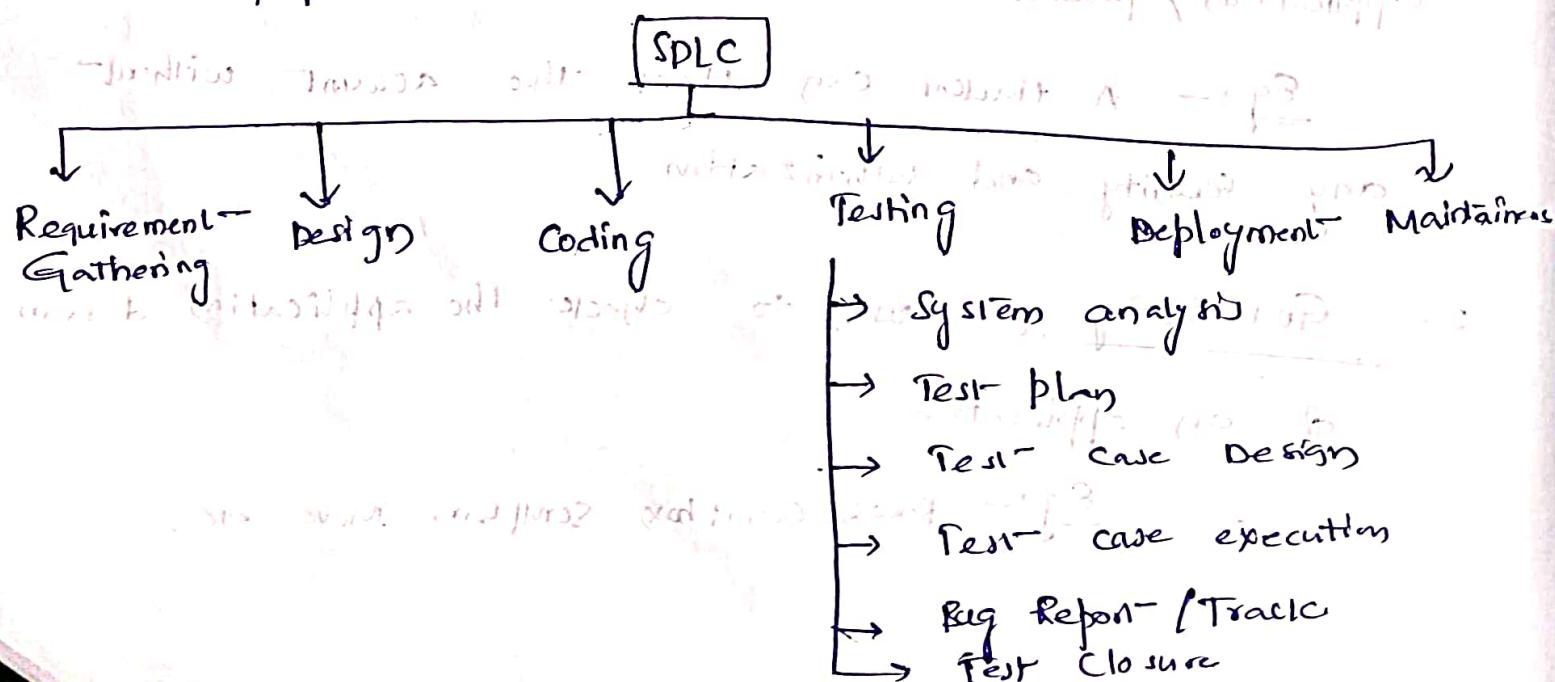
4. Stress testing :- The testing used to determine the stability of an given system.

5. Load testing :- It is the process to determine a system behaviour under both normal and at peak conditions.

6. Performance testing :- A performance testing is testing that is performed, to determine how fast some aspect of system performs under a particular workload.

Q) Draw and Explain software testing life cycle in detail.

a) SPLC :- Software testing life cycle is an systematic approach to testing an software application to ensure that it meets the requirements and it free of defects. It is a process that follows series of steps/phases, each phase has its specific objectives and deliverables.



① System Analysis :- It means what software we are developing, what are client requirements, what are we developing, what are client requirements, what are we need to test and develop are analyzed in this phase.

② Test plan :- This is done by senior in the team. In these phase they make plan how to test, what to test, how many sprints to test, how many user stories are available, too many feature functionalities, which team involves in which period / work all done in this phase.

③ Test Case Design :- The software testing write the test cases. They can be written in excel sheet / Test Management System. ~~Companies~~ ~~Instructions~~ ~~Test cases~~ contain instructions. The instructions are used to test the features, functionalities are to be tested by what manner.

④ Test Execution :- It is used to execute the test case and check whether the features of an application are meeting the client requirements / not.

⑤ Bug Report / Trace :- It is the process of reporting the all data about the error occurring in Client and product Requirement output. There is an deviation between expectations and actual output.

6. Test closure - The test cases are all passed and no more issues are expected and diverged output then the team make a decision (meeting then forward to client and close).

Q) Explain the different types of defects during ST?

1. Functional defect
2. Unit level defect
3. Integration level defect
4. Usability defect
5. Performance defect
6. Security defect
7. Compatibility defect

Answers are based on priority and severity of defect

Q) Write short notes on priority and severity of defect

A) Priority is a term that defines how fast we need to fix a defect.

→ A high priority defect needs to be fixed as soon as possible.

Priority can be following types of

1. High :- A defect with high priority must be resolved as soon as possible because the defect is affecting the application or the product severely usually all high security bugs are high priority

2. Medium :- These are solved after high priority bugs resolved

3. Low :- These are solved after medium bugs resolved.

Priority based on criticality of defect

Defect - Severity

→ This stands for the degree of impact that defect can have on testing and user needs.

High :- Showstopper, tester/user cannot proceed further

Eg:- Cannot login, system hangs, crashes, closes abruptly, corrupts data in database

Medium :- work-around an alternative is available.

Eg:- Entering from front door is not possible then alternative backdoor is possible

Low :- cosmetic, spelling mistakes etc.,

Q) Explain briefly 7 principles of software testing

1. Testing shows presence of defects
not absence of bugs helps in finding undiscovered defects

2. Early Testing :- collaboration with engineers
should start testing as soon as possible.

Finding defects early can save money rather than finding later.

3. Testing - context Dependence :-

Testing is context dependent.

Different websites are tested differently

Eg:- Banking tested different from shopping site

4. Exhaustive testing is impossible
impossible to test all possible input combinations of data and scenarios. Smarter ways of testing should be adopted.

5. Detect clustering :- equal distribution of bugs across the modules is not possible. Defect may be clustered in small piece of code / module.

6. pesticide paradox :- executing same test cases again and again will not help to identify more bugs. Review them regularly and modify if changes required.

7. Absence of errors fallacy: This is another important aspect of it.
Finding and fixing many bugs does not help if it fails to meet user's requirements. It is not useful if it does not support what the user wants.

A) Test scenarios: - It is any condition or functionality that can be tested. It is also called test condition or test setup.

Test possibility
that sets in parallel blind
to the blind
and with which the same set

Test Cases: It can be a document that contains the steps that has to be executed if has been planned earlier.

Test script: It is written in a programming language and it's a short program used to test part of functionality of software systems. In other words a written set of steps that should be performed manually.

Requirement Traceability Matrix (RTM)

1. Define RTM and some of its parts:
 - A) A Requirements traceability Matrix is a tool that helps identify and maintain the status of the project's requirements and deliverables.
→ It does so by establishing a thread for each component.
→ It also manages the overall project requirements.

2. What is build?

- A) In software Testing, a "build" refers to a version of the software that has been compiled and packaged for testing.
→ Build testing is the process of testing each build of the software to ensure that it meets the necessary quality standards before it releases.

3. what is the difference between Regression testing and Reverting

Regression testing	Reverting
1. It is called as generic testing	1. It is called as planned testing
2. Used for passed test cases	2. Failed test cases
3. Defect verification not come under Regression testing	2. Defect verification comes under Reverting

4. Explain differences between smoke testing and sanity testing?

Smoke testing	Sanity testing
1. Done to assure that the acute functionality of program is working fine	1. Done to check bugs have been fixed after the build
2. Called as subset of acceptance testing	2. Called as subset of Regression testing
3. It is not documented	3. It is not documented
4. performed by developer	4. performed by tester

5. outline the differences between adhoc testing and exploratory testing.

<u>Adhoc testing</u>	<u>Exploratory testing</u>
1. Target or goals are not set	1. Target or goals are set
2. There is no documentation	2. There is an proper documentation
3. It is little practical	3. good practical
4. Testing done without any predefined test-plan or test-cases	4. Testing done on the basis of knowledge, experience and intuition

6. write the difference between load stress and stress testing

<u>Test</u> <u>load stress</u>	<u>Test</u> <u>stress test</u>
1. Used by networks, database and website servers to determine application performance	1. The Response time and reliability of the system was checked by stress.
2. load on every application or software has reproduced this type of testing	2. It is used to access the application reliability and robustness

7. Write a defect having high priority and low severity?

A) A spelling mistake in the website's name on the main page is considered a high priority issue although it does not affect the functionality and can be marked as low severity.

8. List-out Test Deliverables?

- A) 1. Test strategy
2. Test plan and estimation
3. Test scenarios
4. Test case documents
5. Test data
6. RTM
7. Test summary report

9. What is test case?

- A) It is sequence of actions or circumstances that are devised to confirm the effectiveness of a software program or application.
- ⇒ It aims to ensure that the software operates as planned and that all its features and functions perform correctly.

10. Why test cases should be developed for valid and invalid inputs?

- A) Due to erroneous conditions and malfunctions.
- 1 For revealing defects since they may execute the code in unexpected ways and identify unexpected software behaviour.

Unit - I

2M

1) What is walkthrough?

- A) It is an software peer review.
⇒ Used to review documents with peers, managers and fellow team members who are guided by the author of the document to gather feedback and reach a consensus.

2) What is SR?

- A) Specific Requirements - Specification is a document that describes what the software will do and how it will be expected to perform.

3) Which SDLC Model is best and why?

- A) Agile Model
⇒ Combination of iterative and incremental approach focussed on fitting in well with flexible requirement and high Customer collaboration.

4. Explain High level Design

- a) It explains the architecture that would be used to develop a system or provides an overview of entire system, identifying the main components that would be developed for the product.
- b. Explain low level design!
- c) It is detail description of every module of software - it says the logic behind the every module

5. What is FRS document?

- a) Functional Requirements Specification
- It lists out all the functions that a software program or product must accomplish.

6. List out Briefly 5 types of PLC Modeling

- A) 1. Waterfall
2. V-shaped
3. Iterative
4. Spiral
5. Agile

To implement
- first step -
second -
third -
fourth -

7. what is the main purpose of performance testing?

- A) To analyze various success factors such as response time and potential errors.
⇒ with this we can find bugs, errors and mistake

Q. Do you think automated test tools are better than manual testing? If yes, then why?

A. Yes

- Automated testing is more reliable as it is performed by tools and scripts.

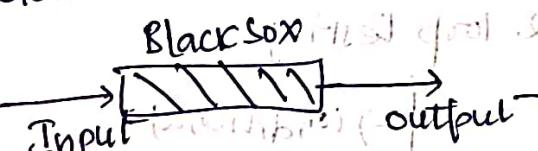
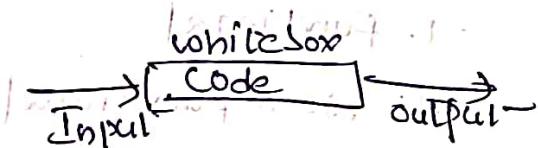
Q. What is difference between functional testing and Usability testing?

Functional	Usability
1. Functional testing is performed to validate if the software is meeting the functional Requirements or not	1. validating how easy to use the app
2. It is practised before Non-Functional testing	2. It is practised after functional testing
3. consists of <ul style="list-style-type: none">- unit test- smoke- Regression- integration	3. It comes under Non-Functional

Software testing

SK. Sardam
Cee-ai!

- Outline the differences between white box testing and black box testing

SD Black Box	WhiteBox
<ul style="list-style-type: none"> → It is high level testing It focuses on the behaviour of the working software. → In this testing, the programming code of the software is hidden and nothing is known about it. <p style="text-align: center;"></p>	<ul style="list-style-type: none"> → It is a technique which checks the internal functions of the system. → In this testing method, the programmer must have knowledge about the internal code. <p style="text-align: center;"></p>
<ul style="list-style-type: none"> → It focuses only on input and output. → It is mostly done by the software tester. 	<ul style="list-style-type: none"> → It focuses on overall software code, testing. → It is mostly done by software developer.
<ul style="list-style-type: none"> → It is referred as outer (or) external software testing. → It is less-time consuming (as it is less effort). 	<ul style="list-style-type: none"> → It is the inner (or) internal software testing. → It is more time-consuming.

⇒ IT is also called as behavioural testing (or) closed testing

⇒ IT is also called as clear box testing (or) glass testing

⇒ NO knowledge of programming, implementation is required

⇒ Mandatory to have the knowledge of programming, implementation is required.

⇒ IT is not well suitable for algorithm testing

⇒ IT is well suitable for algorithm testing

⇒ Types :-

1. Functional

2. Non-functional

⇒ Types

1. path testing

2. loop testing

 → condition

 → control

 → Branch

2) Write a short note on software development life cycle!

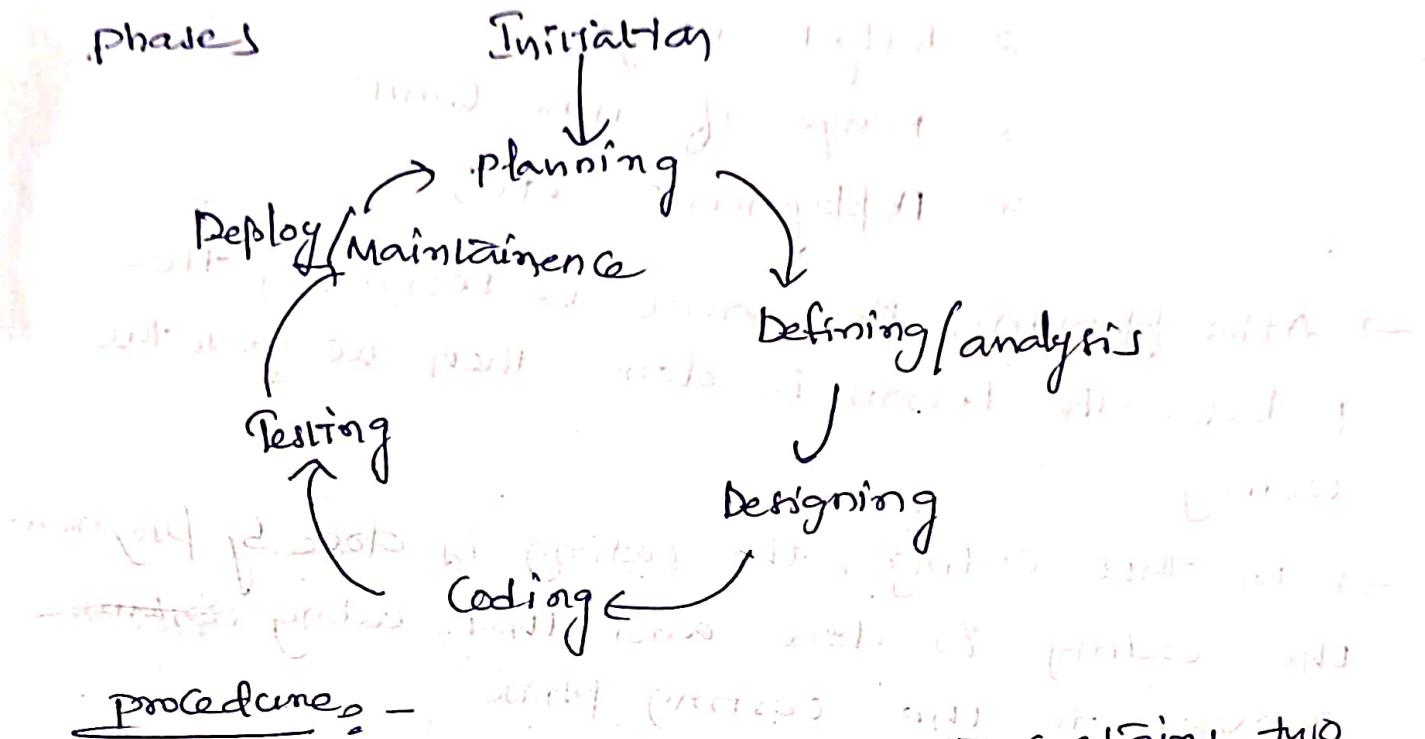
4) Software development life cycle :-

IT is a process of developing any software (client need) in various phases

to get an desired output in a cyclic life. The desired output is the actual output if it meets the expected output.

To meet the expected output we need to

develop the software from bugs, errors etc.
The software development life cycle, has several phases



procedures -
→ The software development contains two entities
1. client
2. service provider

→ The client meets the service provider to give our project. It is called Initiation phase. In these initiation the meeting takes place and gives the requirement, expected output to service provider.

→ The next phase is planning, the software provide service plans how to make the product with the customer satisfaction like in range of time, budget, quality product, desired output

The planning includes

1. Team division
2. Budget - usage
3. Range of time limit
4. Deployment etc.

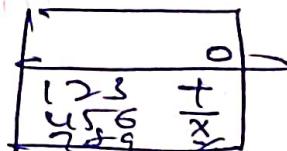
- After planning, the service is designing the product. The design is done then we go to the coding
- In this coding, the coding is done by programer the coding is done and that coding moves to the testing phase
- In testing phase the tester's are test the code. If there is error the send to the code department, else moves to deployment
- In deployment, the actual software is released to market with the expectations of client
- Eg:- calculator software is one of the softwares developed with the help of Java

Step 1:- planning about the requirements

→ Subiect our brain and consider diverse string

Step 2:- After planning design of the calculator

In this project we have to implement simple fo



Step-3:- After design, write a code to perform the activity by calculator. If code is written in modules it will easy to understand.

Step

Eg:- void sum()

Mathematical expression
 $x = a + b$

void sub()

$y = a - b$

void mul()

$z = a \times b$

void div()

$w = a / b$

Step-4:- After coding, move to testing phase if testing is good then the product of calculator software is ready.

Step-5:- In these steps, deployment is done with the expected output.

Ques 3 :- Explain briefly about various SDLC models

A) SDLC :- To develop a product in phases we use SDLC.

SDLC Models

1. Water fall

⇒ Basic model (or) simple model

⇒ First model to develop software

⇒ used for small product

⇒ After completion one phase only if moves to another

⇒ no feedback

⇒ After completion of one phase there is no change in that model

2. Iterative waterfall Model :-

- ⇒ contains feedback
- ⇒ simple model

3. V-Model :-

⇒ extension of waterfall

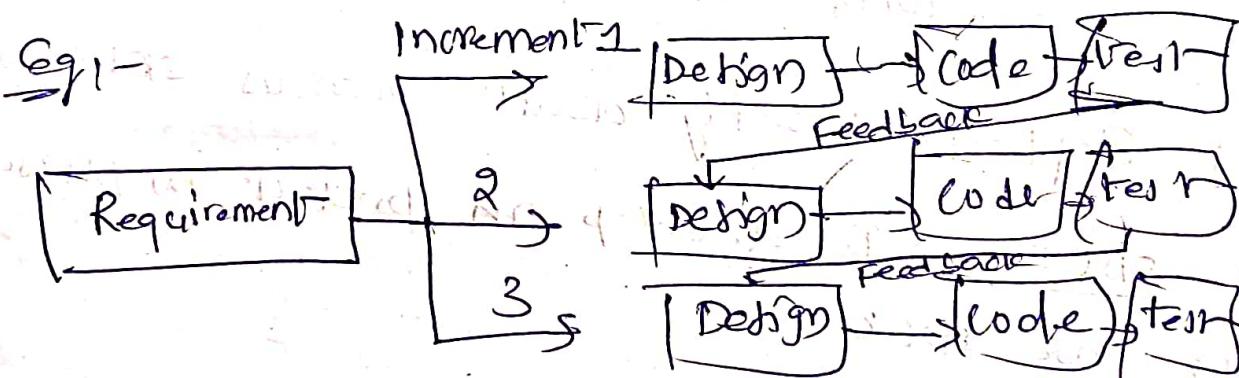
⇒ parallel design and testing takes place

⇒ verification and validation is done at same time

4. Incremental Model :-

⇒ To build large and complex system

⇒ it is "build some, test some, deploy some" mechanism

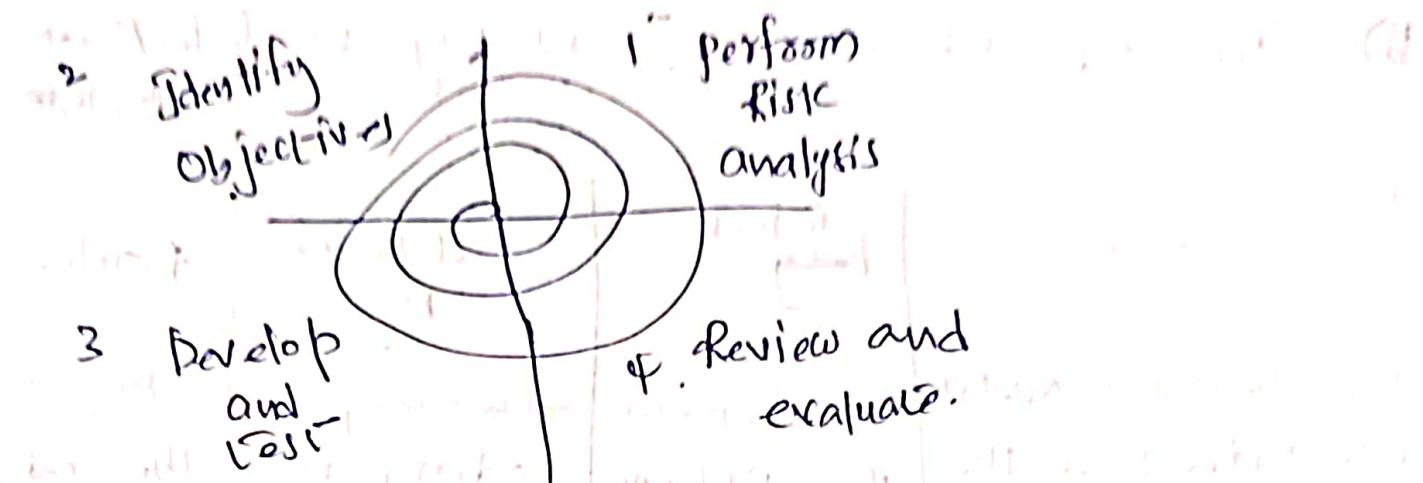


5. Spiral Model :-

⇒ used for Risk Management

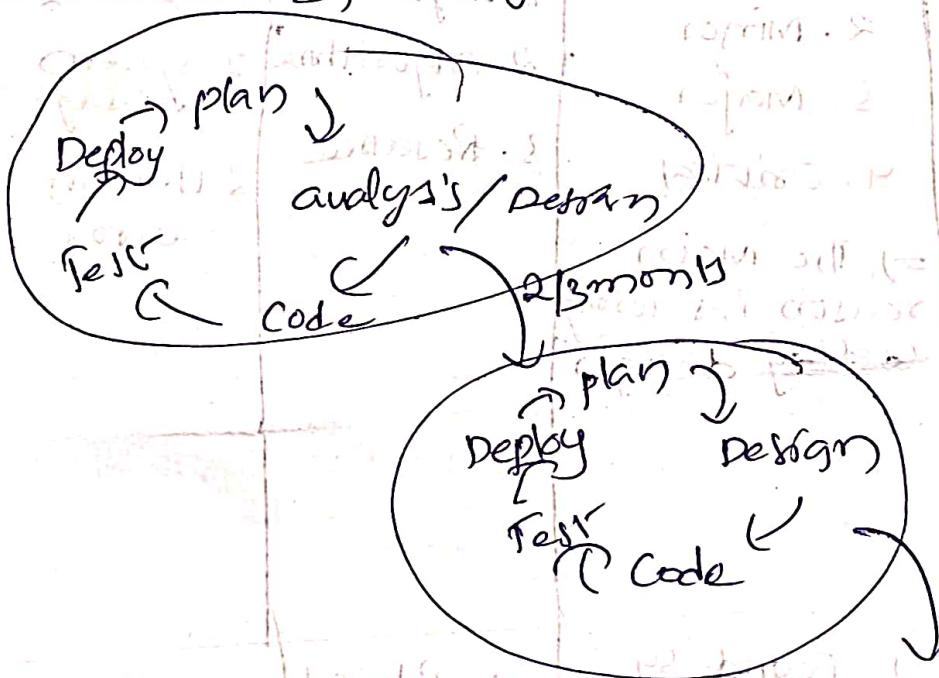
⇒ combination of iterative & waterfall elements

⇒ used for large projects



6. Agile Model

- Contains pair programmers (one to test and one to code and vice versa)
- Every 2/3 months new cycle starts
- Customer collaboration is high
- Large projects



7. RAD Model

- used for small projects
- follows same syntax
- completes within 90 days

Q) difference between Error vs Bug vs defect

Error	Bug	defect	Failure
<p>⇒ problem or mistake identified at the time of development</p> <p>Called error</p>	<p>⇒ Detecting the difference between the actual outcome and expected outcome</p> <p>$\text{actual} - \text{expected}$</p>	<p>⇒ wrong coding is main reason of bugs</p>	<p>⇒ bug at the end of application</p> <p>User error</p> <p>Called Failure</p>
<p>⇒ it is due to unable to compile or run a program successfully</p>			<p>⇒ Found at product</p> <p>Called bug</p>
<p>⇒ Types:-</p> <ol style="list-style-type: none"> 1. syntactic 2. User Interface 3. error handling 4. calculation 5. logical 	<p>Types:-</p> <ol style="list-style-type: none"> 1. Trivial 2. Minor 3. Major 4. Critical <p>⇒ The main reason is wrong Coding design</p>	<p>Types:-</p> <ol style="list-style-type: none"> 1. Logic 2. Algorithmic 3. Resource 	<p>Types:-</p> <ol style="list-style-type: none"> 1. Environment 2. System usage 3. Human error
<p>⇒ Main Reason is improper of syntax (, , ...)</p>			
<p>⇒ It is brought up by developer and automated Test Engineers</p>	<p>⇒ found by Testers, also developer act at the early stage of development</p>	<p>⇒ Repaired by Test Engineers</p>	<p>⇒ During the Development The Manual test engineer discover</p>

4) Verification and validation

i) verification

→ It is process of checking that a software achieves its goal without any bugs.

ii) validation

→ validation is the process of checking whether the software product is up to the mark. It means checking the development is right or not.

→ It includes checking documents, design, codes and programs

iii) static testing

→ It does not include the execution of code

iv) Design phase

→ Methods are
→ Review
→ walk through
→ inspection
→ desk-checking

→ It includes testing and validating the actual product

Dynamic testing

→ It includes Execution of code

Testing phase

→ Methods are
→ Black Box
→ white Box
→ Non-functional

chase from
achase states

Pqr

its equivalent

Ratey

⇒ it can find the bugs at early stage of development

⇒ it can only find the bugs which can't find by verification process

→ Goal is Evolution of application generated by leading software architect and software architecture

→ Goal is improvement in quality of application and actual product

→ verification is about process, standard and guideline

→ it is about product

Project management

Process Standard

Address standard in the standard document

standards for testing

Model oriented

Model oriented

System oriented

System oriented

Test oriented

Test oriented

Code oriented

Code oriented

Requirement oriented

Requirement oriented

Object oriented

Object oriented