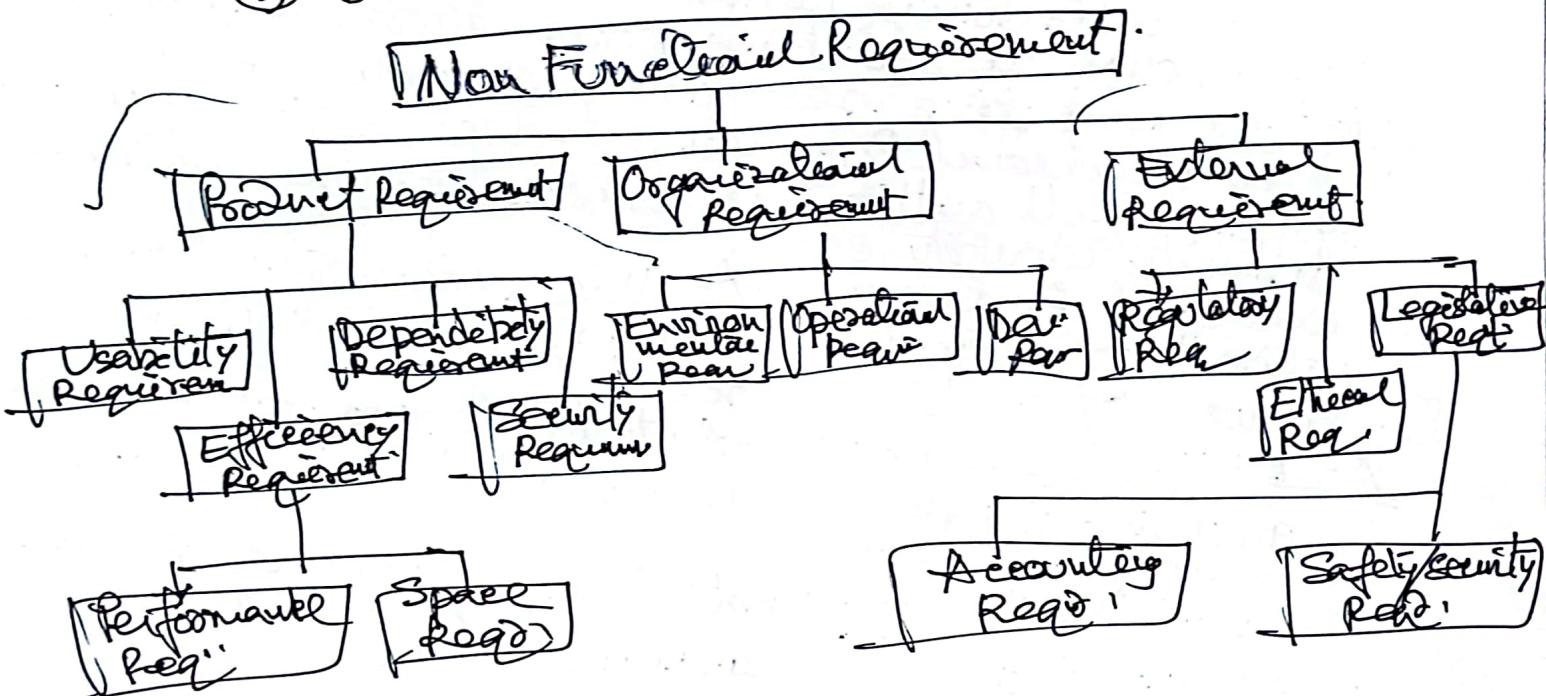


What is Software Requirements:-

A requirement is a feature that the system must have or a constraint that it must satisfy to be accepted by the client.

Q. Types of Requirements

- Ans:-
- ① User requirement.
 - ② System " → Product Requirement.
 - ③ Functional " → Organizational "
 - ④ Non-Functional " → External "
 - ⑤ Domain "



Q. Functional Requirements for the MHC-PMS

- Ans:-
- i) A user shall be able to search the appointment list for all clinics.
 - ii) The system shall generate each day, for each clinic, a list of patients who are expected to attend appointments that day.
 - iii) Each staff member using the system shall be uniquely identified by his or her 8-digit employee number.

Q. Example of Non-functional requirements in the MHC-PMS

Ans:-

- i) Product Requirement - The MHC-PMS shall be available to all clinics during normal working hours (Mon-Fri, 8:30-17:30). Duration within normal working hours shall not exceed five seconds in any one day.
- ii) Organizational Requirement - User of the MHC-PMS system shall authenticate themselves using their health authority identity card.
- iii) External Requirement - The system shall implement patient privacy provisions as set out in HSLan-03-2016-PRB.

Q. Sample of a Test Case:-

Ans:- Title :- Login Page - Authenticate successfully on quact.com.
Description :- A registered user should be able to successfully

Requirement :- A registered user should be able to successfully

Precondition :- The user must already be registered with an email address and password.

Assumption :- A supported browser is being used.

Test Steps:-

- i) Navigate to quact.com.
- ii) In the 'email' field, enter the email of the registered user.
- iii) Click the 'next' button.
- iv) Enter the password of the registered user.
- v) Click 'Sign in'.

Expected Result :- A page displaying the quact user inbox should load, showing any new message at the top of the page.

Q. Test Development Process (A.T.P)

Ans - Test development process includes

> Test Analysis

- Defining the approach.
- Identifying the right techniques.
- Identify associated risks.

> Test Design

Test design involves the creation of test cases and test data.

> Test Implementation

- Test implementation is about defining test procedures to group the test cases in an appropriate way.
- Test implementation also involves preparing test execution schedule.

Q. Identify the test techniques

Ans - External Description

(A) Specification Based (Black Box)

- a) Boundary Value Analysis (BVA).
- b) Decision Table.
- c) State Transition.
- d) Use Case Testing.

- i. Software External Description
- 2. Technical Specification

- 3. Design.

- 4. Customer Requirements.

- 5. Limit & Range.

- 6. Cause & Effect.

- 7. Valid / Invalid.

- 8. Normal / Null payment.

- 9. Marketing / Mill payment.

- 10. Internal software & program.

(B) Structured Based (White Box)

- a) Condition Coverage.
- b) Decision.
- c) Statement.
- d) Multiple Condition.

- 1. Percentage.

- 2. True / False Coverage.

- 3. Requirement Coverage.

- 4. Statement, Component, Function.

- 5. Some statement divided by.

- 6. Component Level.

- 7. Piece of code, Percentage.

- 8. Yes / No.

(C) Experience Based

- ① No formal or limited documentation.
- ② Exploratory Testing Session.
- ③ Test condition, Cases or script.
- ④ Test cases or script.
- ⑤ Fault Attack / Error Guessing.
- ⑥ Random Testing.
- ⑦ Division by zero, Blank input, Empty files, Incorrect Data.

Q. What is the fault attack or error guessing testing? (SOL)

Ans:- One of the widely used methods of the experience-based test design technique, fault attack involves testers anticipating the errors, defect data analysis etc. as per their experience.

Q. Exploratory Testing :- This method generally conducted by business analysts and experts, is followed to test the applications without any documentation.

Q. Experience-Based Test Design Techniques:- The experience-based technique neither involves internal nor the external structure, but is based on experience. Some of the methods follows are -

- i) Exploratory testing
- ii) Fault Attack or error guessing.

Q. Decision Table Testing?

Ans:- Identifying the test conditions as per decision tables which are associated with different conditions is known as decision table testing. Each and every decision keeps up a correspondence to predicates, relations or variables. Decision table which have a 'hyper' symbol are known to have little influence on the actions that are being performed.

Q. Boundary Value Analysis (BVA):-

Ans:- The best method of the black box design technique Boundary value analysis comprises testing the input values at the boundaries. Generally the input values are put to test at the initial stages to reduces of course errors.

Draw a
Software Testing Models :- Today's world technology is completely dominated by machines, and their behavior is controlled by the software power. Software testing provides the solution to all our queries about machines behaving the exact way we want them to. This article will provide in-depth knowledge about the different software testing models in the following sequence: Software Testing Models in the following sequence: Software Testing is an integral part of the software development life cycle.

- ① Waterfall Model.
- ② V-Model.
- ③ Agile Model.
- ④ Spiral Model.
- ⑤ Iterative Model.

Waterfall Model :- This is

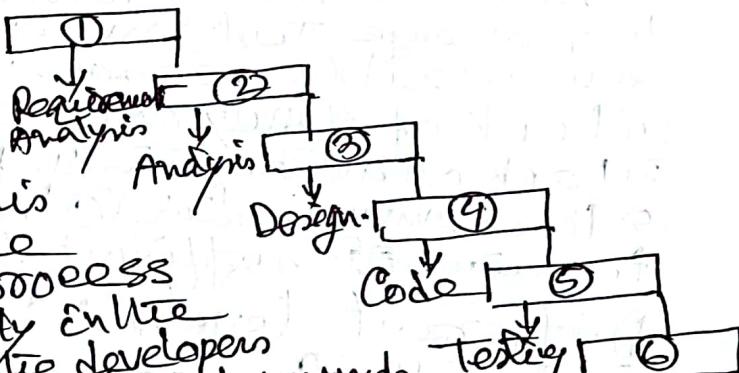
the most basic software development life cycle process which is followed broadly in the industry. In this model, the developers follow a sequence of processes downwards towards the ultimate goal. It is like a waterfall where there are various phases involved (1) Requirement Analysis (2) Analysis (3) Design (4) Implementation (5) Testing (6) Maintenance.

Advantages :- (1) It is easy to implement and maintain.

(2) The initial phase of rigorous scrutiny of requirement and system helps in saving time later in the development phase. (3) The requirement of resources is minimal.

(4) The requirement of resources is minimal.

Disadvantages :- (1) It is not possible to alter or update requirements. (2) Once we move into the next phase we cannot make changes. (3) We can't start the next phase until the previous phase is completed.



Types of Test Environments - The various types of test environments utilized in software development, presented in the order they are typically executed during the Software Development Life Cycle (SDLC).

- 1) Unit Testing Environment.
- 2) System Integration Environment
- 3) Quality Assurance Environment.
- 4) Security Testing Environment
- 5) Performance Testing Env.
- 6) Chaos Testing Environment
- 7) Regression Testing
- 8) Alpha
- 9) Beta Testing
- 10) User Acceptance

Why Companies struggle to manage Test Environments:-
There are many reasons why companies struggle to manage test environments effectively. Hence are a few possible reasons:-

- ① Lack of clarity on ownership
- ② Insufficient Resource
- ③ Lack of automation
- ④ Inadequate testing processes
- ⑤ Poor Communication
- ⑥ Lack of visibility
- ⑦ Insufficient training
- ⑧ Insufficient funding

Different levels & Types of Testing :- There are generally four recognized levels of testing are

- 1) Unit/Component Testing
- 2) Integration Testing
- 3) System Testing
- 4) Acceptance Testing.

Unit/Component Testing

↓
Integration Testing

↓
System Testing

↓
Acceptance Testing

Test are frequently grouped by where they are added in the software development process.

or by the level of the test.

The main Goals of Testing :-

- ① Identifying the defects.
- ② Preventing the defects.
- ③ To check whether the customer requirement is met.
- ④ To measure the quality of the product.

Goals of Software Testing

Risk factors:

- Post
- Time
- Resources
- Critical features

Software Testing

Short term or Immediate Goals.

Bug Discovery

Long Term Goals.

- Reliability
- Quality
- Customer Satisfaction
- Risk Management

Post Implementation Goals.

- Reduced Maintenance Cost
- Improved Testing Process
- Bug prevention

[Software Testing Goals]

[Software Testing] → [Reliability] → [Quality]

(Testing produces Reliability & Quality)

Q. Defect testing is concerned with rooting out undesirable behavior.

System, Review example of such behavior.
A defect is an error or bug in the application which is created. A programmer commits design flaw and which is created. A programmer makes mistakes or errors. building the software can make mistakes or errors. These mistakes or errors means that there are flaws in the software. These are called defects.

Defect testing is concerned with rooting out undesirable system crashes, unwanted interactions with other systems. correct computations and data corruption. We expect the system to perform correctly using a given set of test cases that reflect the system's expected use.

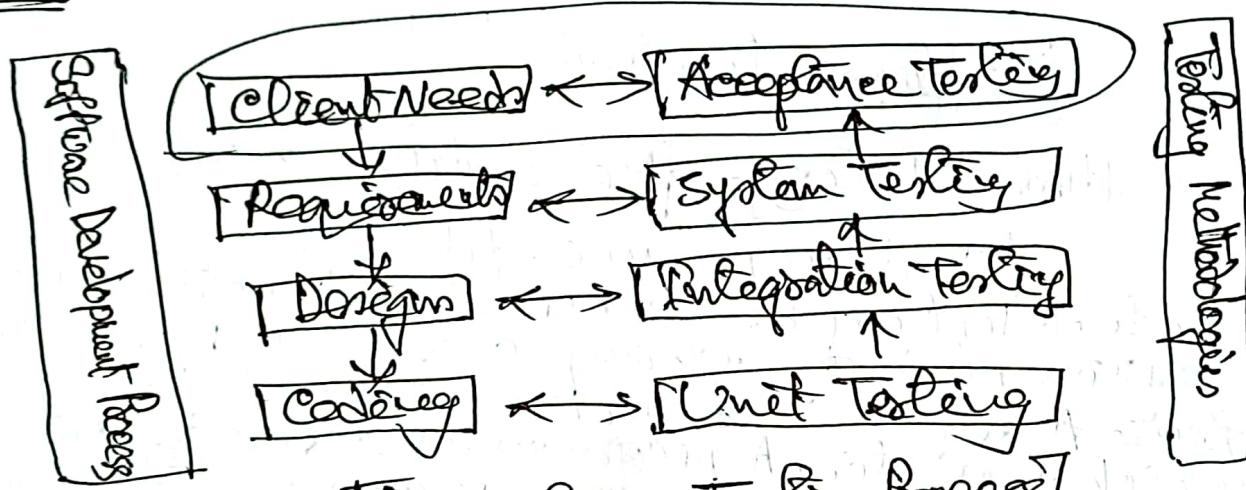
Q. Factors should be considered to write test case for unit testing.

Ans: - For preparing the right test case, a tester should consider the following factors:-

- ① Positive
- ② Negative
- ③ User Interface
- ④ Usability
- ⑤ Performance
- ⑥ Security
- ⑦ Reliability
- ⑧ Compatibility.

Q. Draw the acceptance testing process.

Ans:-



Acceptance Testing Process

Q. What are the stages of testing?

Ans: - According to the test life cycle we follow in our company, the seven stages of testing include test plan, test analysis, test case development, test executing, bug fixing and software implementation.

Test Stages - 1 - Test Plan.

" - 2 - Test Analysis

" - 3 - Test Design

" - 4 - Test Case Development

" - 5 - Test Executing, Bug Fixing and

" - 6 - Bug Fixing and

" - 7 - Software Implementation.

Q. Why testing should be done by developers?

Ans:- During the development process, the developer is required to write down and execute basic test cases which will determine whether the application is structurally sound and performing properly. This is usually done through a process such as unit testing. Software developers lack the objectivity to be able to test their own work. They may find it harder to put themselves into an end user's shoes or assume that the user will have knowledge about something because they do.

The biggest advantages of being a developer is when we switch from one company to another company, the developer role has some extra power and demand than the tester. The hierarchy of a developer is junior developer - senior developer - module leader - Project leader - Project manager.

Developer often focus on technical requirement. Tester often focus on business requirements. Developers can not only build the features during the development phase. Tester finds the defects later in the development process and notify notifying the developers to fix them.

Q. Show the unit testing environment.

Ans:- This type of testing environment is used to test individual source code modules and ensure their accuracy and stability. It is typically used by software engineers to verify that their code is operational and performs as expected.

Various Types of Testing :-

- ① Functional Testing:- This type of testing verifies that the software or hardware functions as expected and meets the specified requirements. It includes testing individual functions, features and interactions to ensure they work correctly.
- ② Regression Testing:- It ensures that new changes or updates to the software do not introduce new defects and that existing functionality remains unaffected.
- ③ Performance Testing:- It evaluates the system's response scalability and stability under various conditions such as load testing, stress testing and endurance testing.
- ④ Compatibility Testing:- It ensures that the software or hardware functions correctly across different platforms, browsers, operating systems and hardware configurations.
- ⑤ Security Testing:- This aims to identify vulnerabilities in the software or hardware and assess its resistance to potential attacks and breaches.
- ⑥ Usability Testing:- It evaluates the user-friendliness and intuitiveness of the software or hardware to ensure a positive user experience.
- ⑦ Acceptance Testing:- It is conducted to determine whether the software or hardware meets the end-users' expectations and satisfies the predefined acceptance criteria.
- ⑧ Smoke Testing:- It is a quick basic test to verify whether the software's critical functionalities are working correctly before performing more extensive testing.
- ⑨ Integration Testing:- It verifies that individual software component or hardware module work together as intended when integrated into a larger system.
- ⑩ Alpha Testing:- It is done internally by the development team to identify bugs and issues before releasing the software.
- ⑪ Beta Testing:- Software to a select group of external users.
- ⑫ Cross-platform and Internationalization Testing:- It ensures that the software can be adapted to different language and cultural contexts.
- ⑬ Recovery Testing:- It assesses how well the software or hardware recovers from various potential situations.

Q. What is driver and stub?

Ans:- A stub is a module or component that acts as a suitable for a called module or component, and provides predefined responses to the caller.

A driver is a module or component that acts as a suitable for a calling module or component, and initiates the calls to the callee.

Q. What are the critical to write driver and stub?

Ans:- The stubs are taken into concern if testing of upper levels of the modules are done and the lower levels of the modules are under developing process. The drivers are taken into concern if testing of lower levels of the modules are done and the upper levels of the modules are under developing process.

Q. What Types of Interface?

Ans:- Types of user interfaces are —

(i) Graphical User Interface (GUI).

(ii) Command Line " (CLI).

(iii) Screen - Driver User "

(iv) Touch user "

(v) Voice User " (VUI).

(vi) Form-based user "

(vii) Natural language " "

Q. What kinds of errors are commonly occurs in the interface?

Ans:- following are the types of interface errors:

(i) Constraint

(ii) Inadequate functionality

(iii) Location of "

(iv) Change in "

(v) Added "

(vi) misuse of Interface

(vii) Inadequate Error Processing.

(viii) Violation of Data Constraints.

(Q9) Defect & Test Management

PMIT 6111: Software Testing & Quality Assurance

Section B, 1st Trimester, Summer 2023

Full Marks: 10

CT#02

Time: 30 Minutes

Answer any two from the following questions

- ✓ 1. What are the criteria for two major classifications of defects? For any application, give examples of following defects:
 - a. High priority and severity
 - b. Low priority and high severity
 - c. High priority and low severity
 - d. Medium priority and medium severity
 - e. Low priority and low severity
- ✓ 2. Write the defect tracking parameters. How any valid defect can be resolved?
- ✓ 3. Mention the factors which influence test estimation. Write the steps of WBS.

Defect Classes

in two perspectives:-

- (i) From the QA team perspective as Priority
- (ii) From the development perspective as Severity.

(iii) From the development perspective (Complexity of code to fix it).

Above the criteria for major two classification of defects.

8. Write the defect tracking parameters.
Ans:- There are various types of defect tracking parameters such as:-

- ① Defect ID.
- ② Priority
- ③ Severity
- ④ Created by
- ⑤ Created Date
- ⑥ Assigned to
- ⑦ Resolved Date
- ⑧ Resolved by
- ⑨ Status.

(1) What is defect? What are the criteria for major & two classification of defects?

Ans:- Defect: - When expected result of test mismatched with actual result of the application, this mismatched is called defect. It can also be error, flaw, failure or fault in a computer program. Most bugs arise from mistakes.

Types of defect: - Following are the common types of defects that occur during development.

- ① Arithmetic
- ② Logical
- ③ Syntax
- ④ Multithreading
- ⑤ Inference
- ⑥ Performance defects.

Defect Classification: - Defects are classified in two perspectives:-

- (i) From the QA team perspective as Priority.
- (ii) From the development perspective as Severity.

(Complexity of code to fix it).

Above the criteria for major two classification of defects.

Q. Write the defect tracking parameters.

Ans :- There are various types of defect tracking parameters such as:-

- ① Defect id.
- ② Priority
- ③ Severity
- ④ Created by
- ⑤ Created Date
- ⑥ Assigned to
- ⑦ Resolved Date
- ⑧ Resolved by
- ⑨ Status.

Q. How any valid defect can be resolved? What is the role of Business Analyst?

Ans - Defect Tracking Process:- After a defect has been found, it must be reported to the development team. So that they can fix the issue.

Any valid defect can be resolved in below steps:-

Step-1:- Tester reported to Business Analysts in project.

Step-2:- If Business Analyst is not available in the project, then reported to co-ordinator for review the defect.

Step-3:- BA or Co-ordinator examine and comment in below

i) the defect is valid

ii) " " is invalid.

iii) " " is duplicate

Step-4:-

i) If the defect is valid, Assigned to development team.

ii) If the defect is not valid, it will be rejected

iii) If the defect is duplicate, its status will be

iv) If the defect is valid, the bugs are resolved

Deferred and identify the bugs to development team.

The Business Analyst role is to examine the defect is valid, invalid or duplicate. After identification, then assigned to development team, if defect is valid or duplicate, otherwise the defect is rejected.

mention the factors which influence test estimation

Ans:- Many external and internal factors impact the project as well as the project timeline. The following factors influenced test estimation:-

- Requirements
- Available Resource.
- Test Data
- Fund Available.

Q. Write the steps of WBS.

Ans- The steps of Work Breakdown Structure (WBS) are below:-

- Step 1:- Create WBS by breaking down the test project into smaller modules.
- "-2:- Divide module into sub-module.
- "-3:- Divide sub-module further into functionalities.
- "-4:- " functionalities into sub-functionalities.
- "-5:- Review the testing requirements.
- "-6:- Figure out the number of tasks.
- "-7:- Estimate the effort of each task.
- "-8:- " duration of " " .

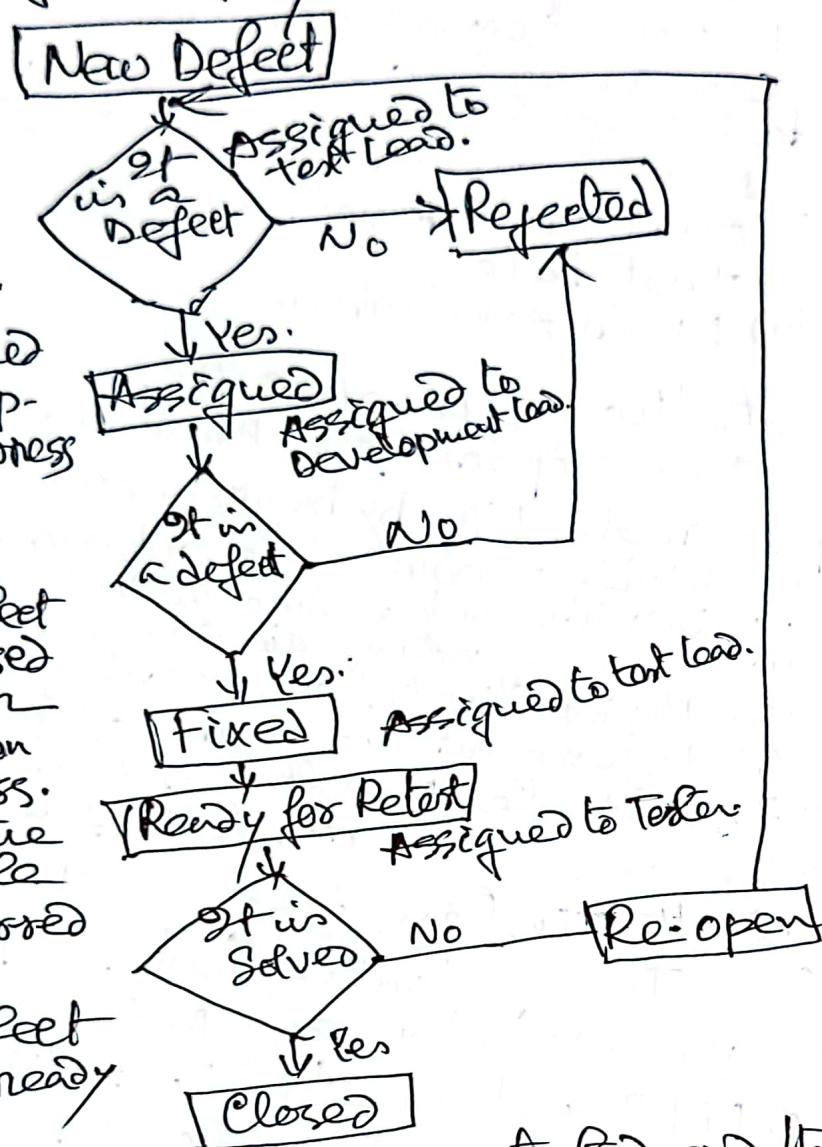
Q. Write the testing Estimation Techniques:

Ans- The following testing estimation techniques are proven to be accurate and are widely used.

- i) PERT software testing estimation technique
- ii) Use Case Point (UCP) method
- iii) Work Breakdown Structure (WBS) method.
- iv) Wideband Delphi technique.
- v) Function point, Testing point analysis.
- vi) Percentage distribution
- vii) Experience-based testing estimation technique.

Q. Draw a defect lifecycle states.

Ans:-



New: - Planned defect that is raised and yet to be validated.

Assigned: - Assigned against a development team to address it but not yet resolved.

Active: - The defect is being addressed by the developer and investigation is under progress. At this stage there are two possible outcome: Deferred or Rejected.

Test: - The defect is fixed and ready for testing.

Verified: - The defect that is tested and the test has been verified by QA.

Closed: - The final state of the defect can be closed after the QA testing or can be closed if the defect is duplicate or considered as NOT a defect.

Reopened: - When the defect is NOT fixed, QA reopens/reactivates the defect.

Deferred: - When a defect can not be addressed in that particular cycle it is deferred to future release.

Rejected: - A defect can be rejected for any of the 3 reasons:- Duplicate defkt, NOT a defkt, Non Reproducible.

For any example application, give example of following defect:-

i) High priority and severity

Example - A spelling mistake on Amazon's home page could be high priority (since it reflects poorly on Amazon as a company) but low severity (since it does not affect the end user).

ii) Low priority and high severity

Example - Informational pages such as the disclaimer and privacy policy are crucial. It may impact other main system functionality. The bug's severity is high, but as many users do not frequently access these pages, it is marked as low priority by the project manager and business team.

iii) High priority and low severity

Example: The company name is misspelled in the footer page of the website. The logo of the company in the footer page is wrong.

iv) Medium priority and medium severity

Example - The inability to go to the next page of the product reviews section on an e-commerce website. Medium severity: - the checkout button not working on an e-commerce website which have been impacted otherwise.

v) Low priority and low severity

Example - The error message displayed is incomplete and contains spelling mistakes.

vi) High priority and high severity

Example - A server error occurs when user tries to access the testing URL.
→ The system crashed after you made the payment.

CHAPTER NO. 05

PMIT 6111: Software Testing & Quality Assurance

1st Trimester, Summer 2023

CT#02: Section A

Time: 40 Minutes

Full Marks: 20

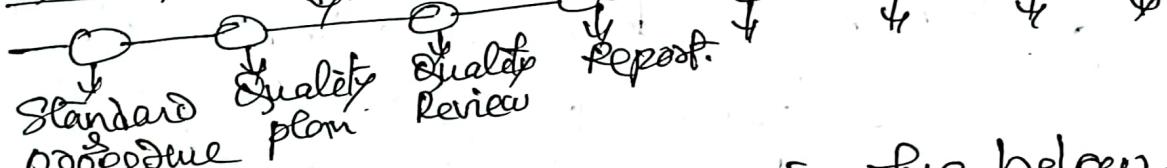
Answer any five from the following questions (4x5=20)

OK

1. Explain quality conflict with an example.
2. Draw the Quality review process. How review can be made easier with agile?
3. How process and product quality depend on each other? Give two examples of product standard and process standard.
4. How agile makes inspection easier? Identify the types of faults:
 - a. If character strings are used, is a delimiter explicitly assigned?
 - b. If a break is required after each case in case statements, has it been included?
 - c. Do formal and actual parameter types match?
 - d. Is space explicitly de-allocated after it is no longer required?
5. How ISO 9001 standard can be achieved by any organization?
6. What is software quality? Mention the software fitness for purpose.

(05) Software Quality Management

Quality management



Software fitness purposes involve below factors

- ① Quality management process.
- ② Standard and procedure.
- ③ Quality Plan.
- ④ Quality Review.
- ⑤ Report.

(05) Software Quality Management Cf1-05'(1)

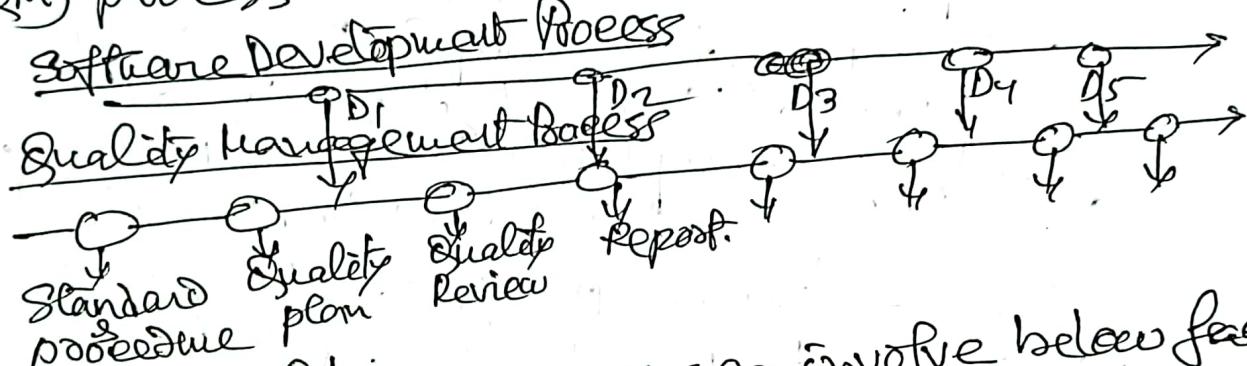
What is software quality? Mention the software fitness for purpose?

Ans:- Software quality is defined as a field of study and practice that describes the desirable attributes of software products. There are two main approaches to software quality:

- i) defect management and
- ii) Quality attributes.

The software fitness for purpose / the software configuration management process mentioned in the
Draw the Quality Software Configuration Management (SCM) process in below:-

(SQM) process in below:-

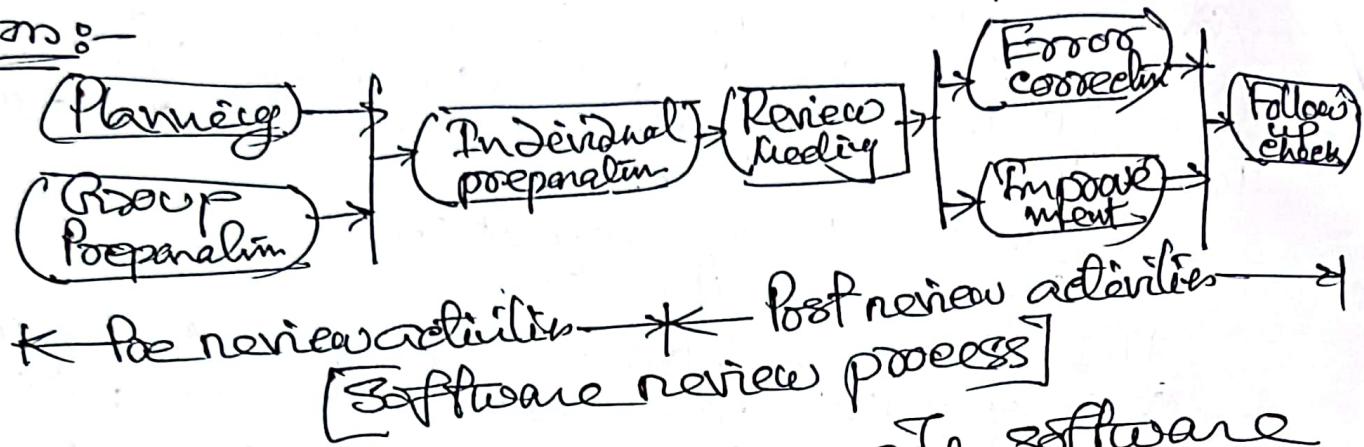


Software fitness purposes involve below facts

- ① Quality management process.
- ② Standard and procedure.
- ③ Quality Plan.
- ④ Quality Review.
- ⑤ Report.

Q. Draw the quality review process? How it can be made easier with agile?

Ans:-



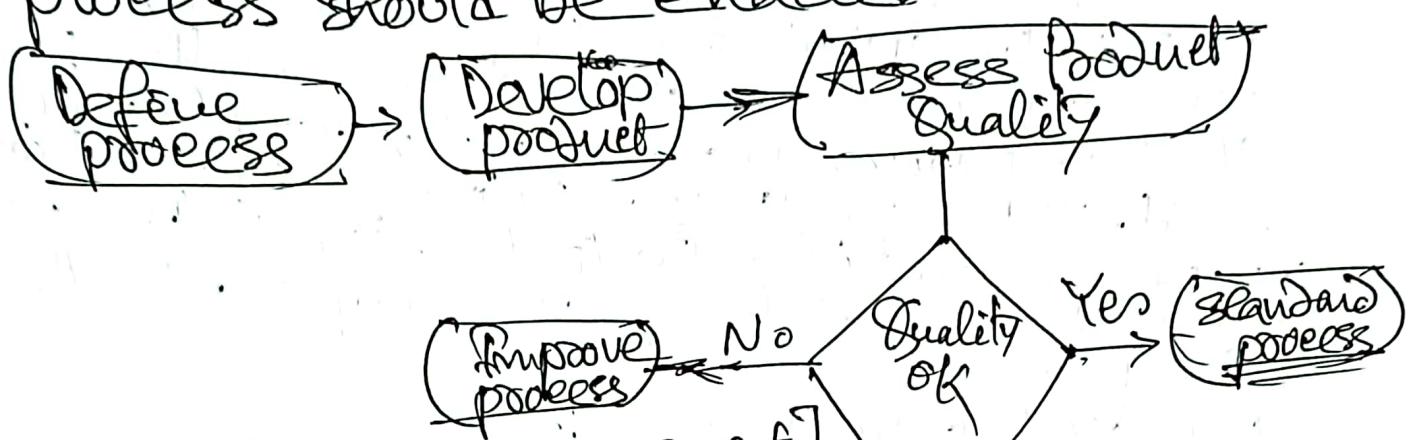
The review process in agile software development is usually informal.

- > In Scrum, for example, there is a review meeting after each iteration of the software has been completed (a sprint review) where quality issues and problems may be discussed.
- > In extreme programming, pair programming ensures that code is constantly being examined and reviewed by another team member.

How process and product quality depends on each other? Give two examples of product standard and process standard. ③

Ans:- Product standard defines characteristics that all software component should exhibit e.g. a common programming style.

Process standard define how the software process should be enacted.



Process Based Quality
The quality of a developed product is influenced by the quality of the production process. This is the important in software development as some product quality attributes are hard to assess. However, there is a very complex and poorly understood relationship between software process and product quality.

Product Standard

- ① Design review form.
- ② Requirement document structure
- ③ Meeting leader format
- ④ Java programming style.
- ⑤ Project plan format
- ⑥ Change request form

Process Standard

- ① Design review conduct.
- ② Submission of new code & build
- ③ Version release process.
- ④ Project plan approval process.
- ⑤ Change control process.
- ⑥ Test recording process.

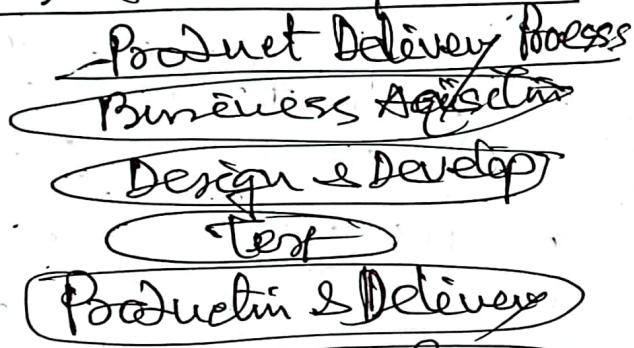
Q: If the ISO 9001 standard can be achieved by any organization. There are below requirement to achieve ISO 9001 certification:-

- i) Quality Management system (QMS)
- ii) Management Responsibility
- iii) Resource management
- iv) Product Realization

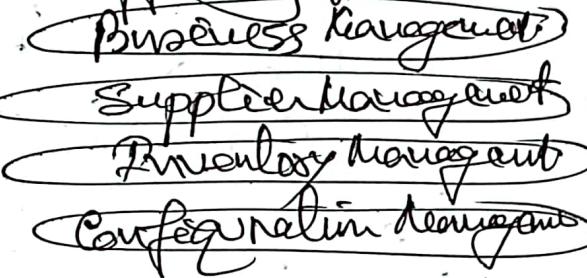
v) Measurement, Analysis, and Improvement

If above mention requirement fulfill, then need to be invited a certification body for audit company performance based on the ISO 9001. If company successfully completed their audit. Then the International certification body issue a certificate that acknowledge company office ISO 9001 status for valid up to next three or five years.

ISO 9001 core process:-



Supporting process.



- c. Quality standard and procedure should be defined in an organizational quality manual.
- iii. An external Body may certify that an organization quality manual conform to ISO 9001 standard.
- iii. Some customer require supplier to be ISO 9001 certified although the need for flexibility here is decreasingly recognised.

Explain quality conflict with an example.

- Ans :-
- i) It is not possible for any system to be optimised for all of these attribute.
 - ii) The quality plan should therefore define the most important quality attribute for the software that is being developed.
 - iii) The plan should also include a definition of the quality assessment process, an agreed way of assessing whether some quality, such as maintainability or robustness, is present on the product.

Q. Problem for quality management software?

- Ans :-
- i) There is a tension between customer quality requirements and developer quality requirements.
 - ii) Some quality requirements are difficult to specify in an unambiguous way.
 - iii) Software specification are usually incomplete and often inconsistent.

Q. Quality Management Activity

- Ans :-
- i) Quality management provides an independent check on the software development process.
 - ii) The quality management process checks the project deliverables to ensure that they are consistent with organizational standards and goals.
 - iii) The quality team should be independent from the development team so that they can take an objective view of the software. This allows them to report on software quality without being influenced by software development issues.

- Q. How agile makes inspection easier? Identify types of fault.
- i) If character strings are used, as a delimiter, a city assigned — Data fault.
 - ii) If a break is required after each case in case statements, has it been included? — Control fault.
 - iii) Do formal and actual parameter types match? — Type fault.
 - iv) Is space explicitly de-allocated after it is no longer required? — Storage management fault.

CH-06.

1. Draw or Show the Risk management Process.
2. Give an example of different kind of risk.

Ans:- Risk management is concern with identifying risks and drawing up plans to minimise their effect on a project.

The risk management process drawn in below figure:-



[The Risk Management Process]

The Risk Management Process with example:-

there are different types of risk with example:-

- (A) Technological Risk: - ① The database used in the system cannot process as many command per second as expected.
② Reusable software component contained that means they can not be reused as planned.
- (B) People Risk: - ③ It is impossible to recruit staff still required.
④ Key staff are ill and unavailable at critical times.
⑤ Required training for staff is not available.
- (C) Organisational Risks: - ⑥ The organization is restricted so that different management are responsible for the project.
⑦ Organizational financial problems force reduction budget.
⑧ The code generated by software code.
- (D) Tools Risk: - ⑨ The code generated by generation tools is inefficient.
⑩ Software tools cannot work together in an integrated way.
- (E) Requirement Risk: - ⑪ Change the requirement that require major design rework are proposed.
⑫ Customer fail to understand the impact of requirement change
⑬ Customer fail to understand the time required to develop the software.
- (F) Estimation Risk: - ⑭ The rate of defect repair is underestimated.
⑮ The size of the software is underestimated.

Q. Identify the types of Risk, probability and effects of the Risk.

Ans:-

SP	Type of Risk	Probability	Effect
01.	Organizational financial problem for reductions in the project budget	Low	Critical
02.	It is impossible to recruit staff with the skill required for the project	High	"
03.	Faults in reusable software component have taken serious to be repaired before these component are reused	Moderate	Serious.
04.	Change to requirement that require major design rework are proposed	"	"
05.	The organization is restructuring so that high different management are responsible	"	"
06.	The database used in the system cannot process as many transaction per second as expected.	Moderate	"
07.	The time required to develop the software is underestimated	High	"
08.	Software tools cannot be integrated	"	Tolerable.
09.	Customer fail to understand the impact of requirement changes	Moderate	"
10.	Required training for staff is not part	"	"
11.	The rate of defect repair is underestimated	"	"
12.	The size of software is	"	High
13.	Code generated by code generation tools	Moderate	Insignificant error
14.	Key staff are ill at critical time in the project	"	Serious.

SOA
P2

Software Quality Assurance (SQA):- SQA is a process which assures that all software engineering processes, methods, activities and work items are monitored and comply against the defined standard. These defined standards could be one or a combination of any like ISO9000, CMMI model, ISO15504, etc. SQA incorporates all software development process starting from defining requirements to coding until release. Its prime goal is to ensure quality.

Elements of SQA:-

- i. Standard, ii. Review & Audit iii. Testing iv. Defect analysis v. Change management vi. Education vii. Vendor management viii. Security management ix. Safety x. Risk management.

SQA Goals:-

- i. Requirement Quality ii. Design quality iii. Code quality
- iv. Quality control effectiveness.

SQA Plan:-

- i. Management plan section
- ii. Documentation section
- iii. Standards, practice & convention section
- iv. Review and audit section
- v. Test section
- vi. Problem reporting & corrective action section.
- vii. Others

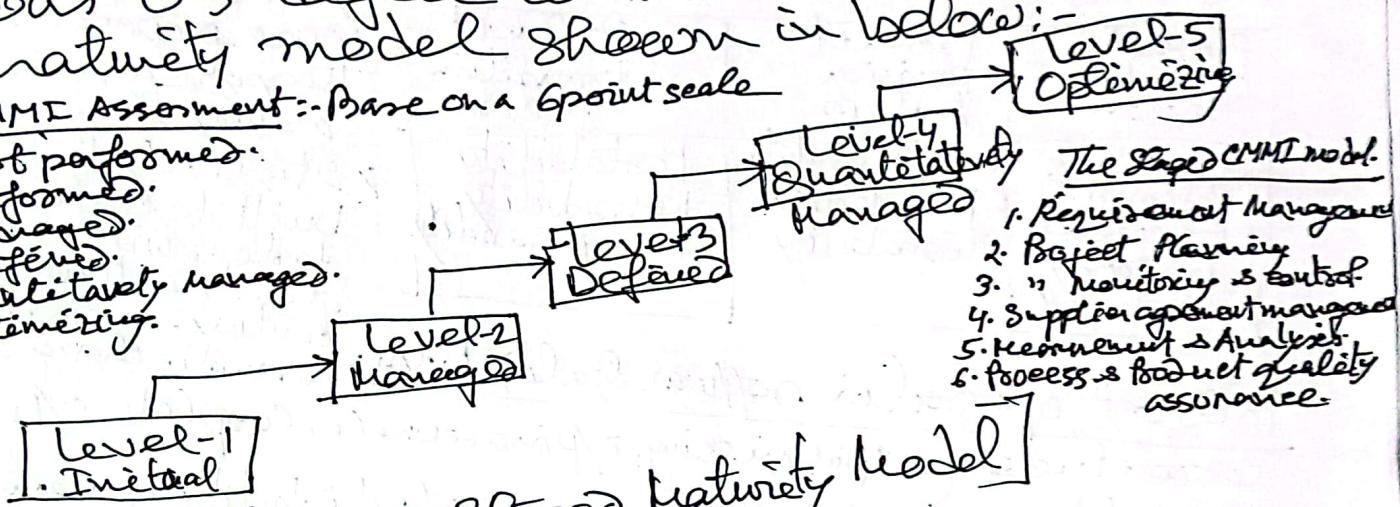
CH-04-

Show the stages of CMMI.

Ans:- The Capacity maturity model Indicator (CMMI) framework is the current stage of work on process assessment and improvement that started at the Software Engineering Institute in 1980's. It was US defense contractor. The CMMI staged maturity model shown in below:

CMMI Assessment :- Base on a 6 point scale

- i. Not performed.
- ii. Performed.
- iii. Managed.
- iv. Defined.
- v. Quantitatively managed.
- vi. Optimizing.



[CMMI Staged Maturity Model]

Q. What is the continuous CMMI.

Ans:- This is a finer-grain model that considers individual or group of practices and assesses them. Here the maturity assessment is not a single value but is a set of values showing the organization's maturity in each area.

The CMMI rates each process area from level 1-5.

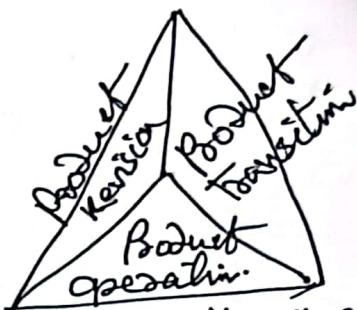
The advantages of a continuous approach is that organization can pick and choose process areas to improve according to its local need.



MACEALL'S Factor Model:-

- > According to this model all software requirements are classified into 11 quality factors.
- > All quality factors are classified again into three groups:-

Product operation factors	Product revision factors	Product transition factors
Correctness Reliability Efficiency Integrity Usability	Reliability Flexibility Testability	Portability Reweability Interoperability



> McCall's quality factors were proposed in the 1970s. They are as valid today as they were in that time. > It is likely that software built to conform to these factors will exhibit high quality well into the 21st century, even if there are dramatic changes in technology.

Product operation software Quality Factors:-

Correctness (output missing, spinaceme, complete op, Up to date)

Reliability: A simple measure of reliability is mean time between failure (MTBF), where:

$$MTBF = MTTF + MTTR$$
 (where $MTTF = \text{mean time to failure}$ & $MTTR = \text{mean time to repair}$)

$$\text{Availability} = \frac{MTTF}{MTTF+MTTR} \times 100\%$$

Software Quality Metrics:-

- 1) Product Quality Metrics :- Problems per User-Month (PUM)

$$PUM = \frac{\text{Total problems that customer reported (true defect and non-defect oriented problem)}}{\text{Total number of licensed months of the software}} \times 100\%$$
- 2) In-process Quality Metrics :- Defect removal effectiveness (DRE).

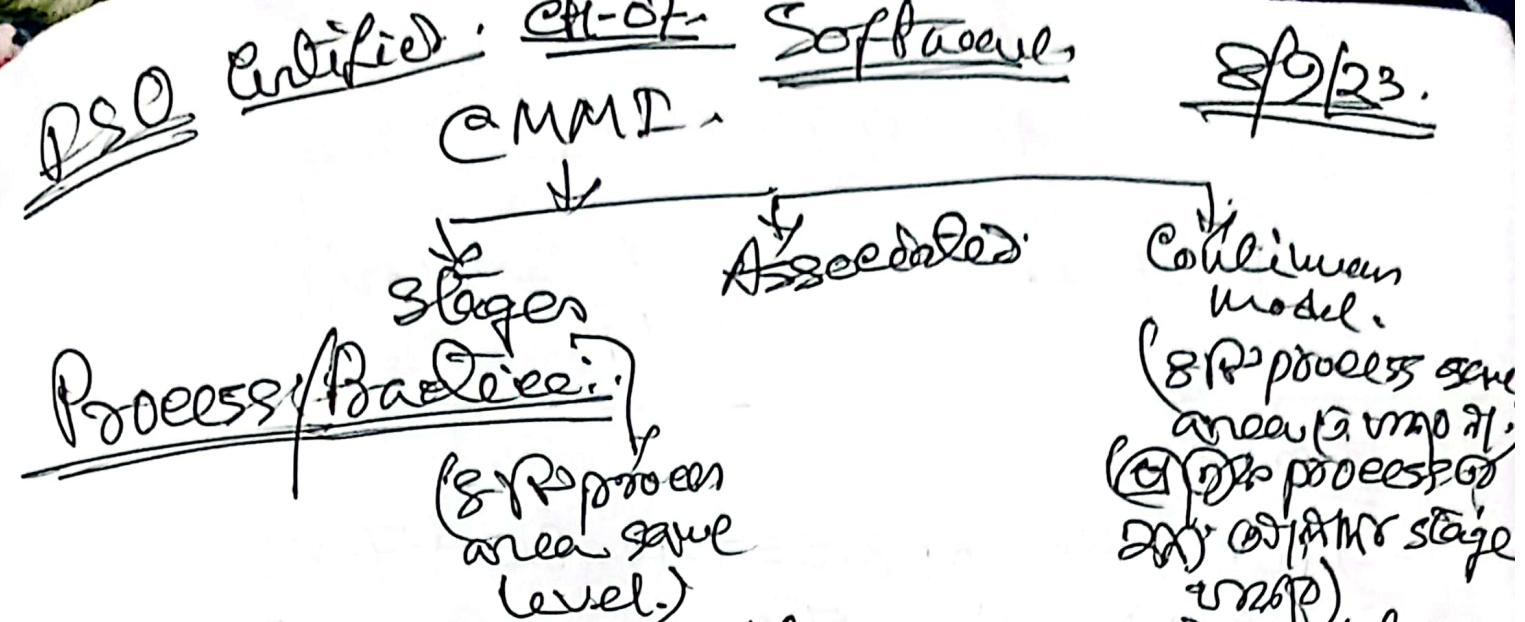
$$DRE = \frac{\text{Defect removal during a development phase}}{\text{Defect latent in the product}} \times 100\%$$
- 3) Maintenance Quality Metrics :- Backlog management index (BMI)

$$BMI = \frac{\text{No. of problems closed during the month}}{\text{No. of problems opened during the month}} \times 100\%$$

Fix response time metric is usually calculated as the meantime fix of all problems from open to close. Short fix response time leads to customer satisfaction.

Percent defective fixes = No. of fixes that exceed the response time criteria of severity level / No. of fixes delivered in a specified time

Fix Quality :-



Quality Factors model:

~~MADE~~ MCCALL'S Factors Model.

Quality factor (S) Process factor
 Quality factor (P) Product factor
 Quality factor (R) Resource factor

→ R model → P & R

Reliability: $MTBF = MTTF + MTTR$. {close to zero as accepted.}

Availability: $\frac{MTTF}{MTTF + MTTR} \times 100$ ~ more is ..

[So system is more reliable]

Quality Metrics

PUM = Problems per User Month.
 = Total problems that customers reported (true defect and non defect created problem) for a time period + Total number of license month of the software during the period.

$$= (70 \times 3) + \underline{\underline{800}} = 210 + 800 = \underline{\underline{1010}}$$

Defect Density = $\rightarrow D.D = \frac{54}{40}$

$$\underline{\underline{800}} = \frac{\text{true def} + \text{non-defect}}{540} = \frac{260}{540} = \frac{456}{540} \times 100\%$$

$$B.M.D = \frac{456}{540} \times 100\%$$

Continuous model.
 (S) Process area group 1.
 (P) Process group 2
 (R) Customer stage group 3.

The model more practical.

(A process expand profile to fit into S)

$$540 - 444 = 104$$

$$D. Fixes = \frac{104}{444} \times 100\%$$

③

Ans :-

$$i) PUM = \cancel{70 \times 3 + 800} - 104 = 800 + 70 \times 3 = 1010$$

$$ii) \text{Defect Density} = \frac{540}{40} = 13.5 \quad (\text{D.D.} = \frac{\text{True Defect}}{\text{FP}})$$

$$iii) BMI = \frac{456}{540} \times 100 = \left(\text{BMI} = \frac{\text{No. of problem closed}}{\text{No. of problem arrived}} \times 100 \right)$$

we know,

$$iv) D. fix = \frac{104}{444} \times 100 =$$

$$\boxed{\text{where, } 540 - 444 = 104}$$

Reported Problem = True Defect
+ Non defect.

$$\Rightarrow 840 = 540 + \text{Non defect}$$

$$\Rightarrow \text{Non Defect} = 840 - 540 = 260$$

5)

$$i) PUM = 50 \times 3 + 200 = 350$$

$$ii) \text{Defect Density}$$