

SOLUTIONS

1. a)

201.

Q) Eliciting requirements means to collect the different requirements and needs and documenting it so that it can be used in developing the product.

The techniques to elicit requirements from various stakeholders for a custom software solution are:-

i) conducting Meetings: The meeting should be conducted frequently in order to communicate the needs and requirements in the software. The questions to be asked are:-

- Who is behind the request?
- What is the need?
- How much resource is required?

ii) surveys: conducting survey helps to get the answer of the actual need of the software. It also clears what type of modification do they want.

iii) Make plans and functions: Based on the initial few meetings and survey, a base plan is made. After that the plan is again discussed with the stakeholders and choose the appropriate plan.

iv) Define constraints: Different constraints should be defined in order to achieve the quality software in the desired time.

v) Prioritize: Prioritize the tasks and execute them accordingly.

vi) use-case: A sample use-case diagram is created so that it will have a brief idea how that software works.

1.b)

During validating the software requirements, the following checklist should be taken care :-

- i) Is each requirement consistent with the project requirements?
- ii) Is each requirement has a valid source?
- iii) Is each requirement conflicts with the existing project requirement?
- iv) Is requirement is relevant to the project plan?
- v) Is requirement consistent with project?
- vi) Is requirement help in achieving the desired feature in the project?
- vii) Is each requirement increases the ^{overall} performance of the project?

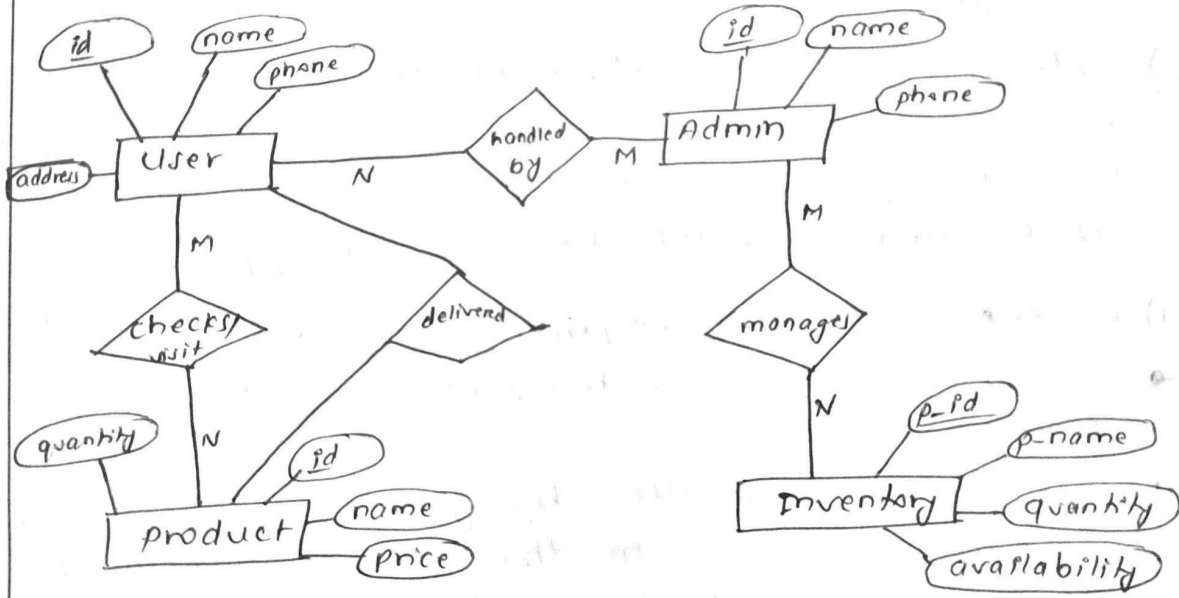
2. a)

→ Data model is a visual representation of the different object or entities and attributes.

→ ~~For~~ In order to create a data model for an online retail store, first we have to identify the entity and the attributes of each entity.

→ After that ~~to~~ what is the relationship between the entities.

ER-Diagram for an online retail store.



2.b.

gn02.

b) ⇒ The different types of software projects are:-

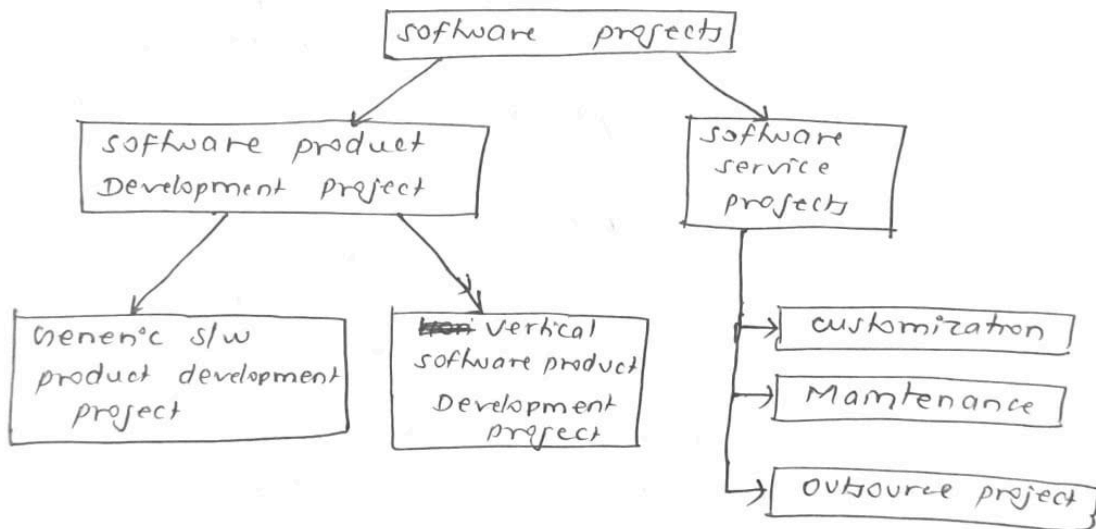


Fig:- Types of software projects

→ The software projects are broadly classified into two types:

1) Software Product Development project:

→ In this project, the software product are developed for the general customers.

→ It is further divided into two sub-types:-

i) Generic software Development Project:

• It is also known as horizontal SW development project.

• Here, the software used by general customer are developed and sell in the market directly as the general customers for this software are very large in number.

• Eg: Microsoft 365, Windows.

ii) Vertical software Development Project.

• Here the software product are developed for the specific fields like medical, engineering, traffic.

• Eg:- AutoCAD for engineering, Financial SW for Bank.

2) Software service product.

→ Here instead of developing a new software, the services are provided for the existing software

→ They are further classified as:

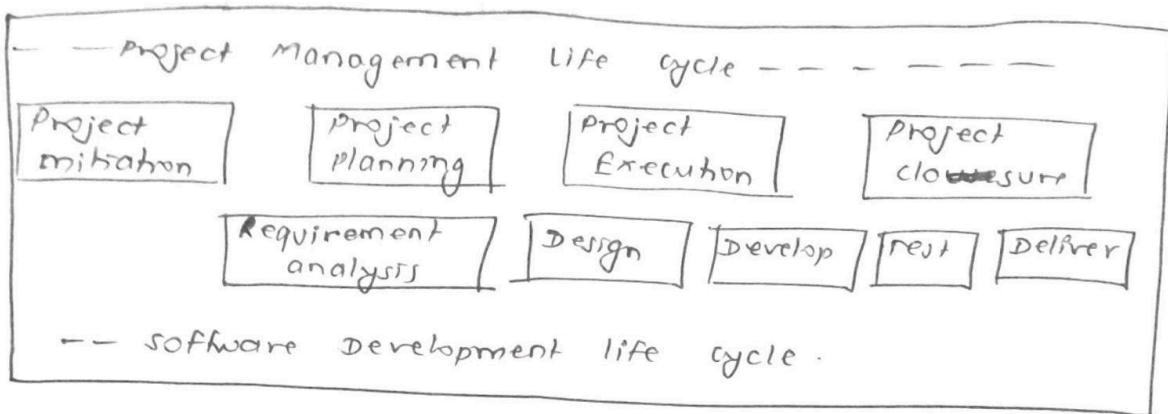
i) Software Maintenance: If any problem arise in the existing working software then, it will fix the issue.

- ii) software customization: In the existing software if someone wants the software with its own customizable features, settings, then it is achieved through this.
- iii) outsource project: If the software cannot be built or developed within an organization then it will be given to third party as outsource project.

3.a

3

→ The different stages in the Project Management Life Cycle are:-



i) Project Initiation:

In this step, the following questions should be answered.

- Who request the project?
- Where is the organization located?
- What is the need?
- When is the submission?
- How to achieve that goal?
- How much resource is required?

ii) Project Bidding-

→ Here, a project charter is created after the business approval.

→ These three methods are followed.

- 1) RFQ → Request for Quotation.
- ii) RFP → Request for Proposal.
- iii) RFI → Request for Information.

iii) Project planning

→ Here, the detail plan is documented by mentioning scope, risk management

→ It is also planned how to allocate the resources and whom to allocate.

iv) Project Execution.

→ In this stage, the actual project is started.

→ First the required manpower is acquired and hand over the task.

→ After that, time to time update is achieved and the deliverable components are delivered to customer.

v) Project closure

→ After the delivery of all the components and also user accepts all the components then the project requirements are handed over to customer.

→ Remaining payment is cleared with organization and the termination of the contract.

3.b)

Boehm's software Quality Model.

→ It is one of the model which defines the software quality.

→ software quality is defined using 3 different characteristics and they are:-

1) Portability: It determines how easy is it to develop or transfer the software to different environment.

It is further divided into 2 category:-

- Device Independence
- self containedness.

ii) As-in-utility: It defines and states how easy is to use the software. The reliability of the software is tested. It is further categorized into 3 types:-

- Reliability
- Efficiency
- Helping Engineering

iii) Manageability: It states that how easy is it to manage the software. It is further divided into 3 categories:-

- Testability
- Understandability
- Manageability

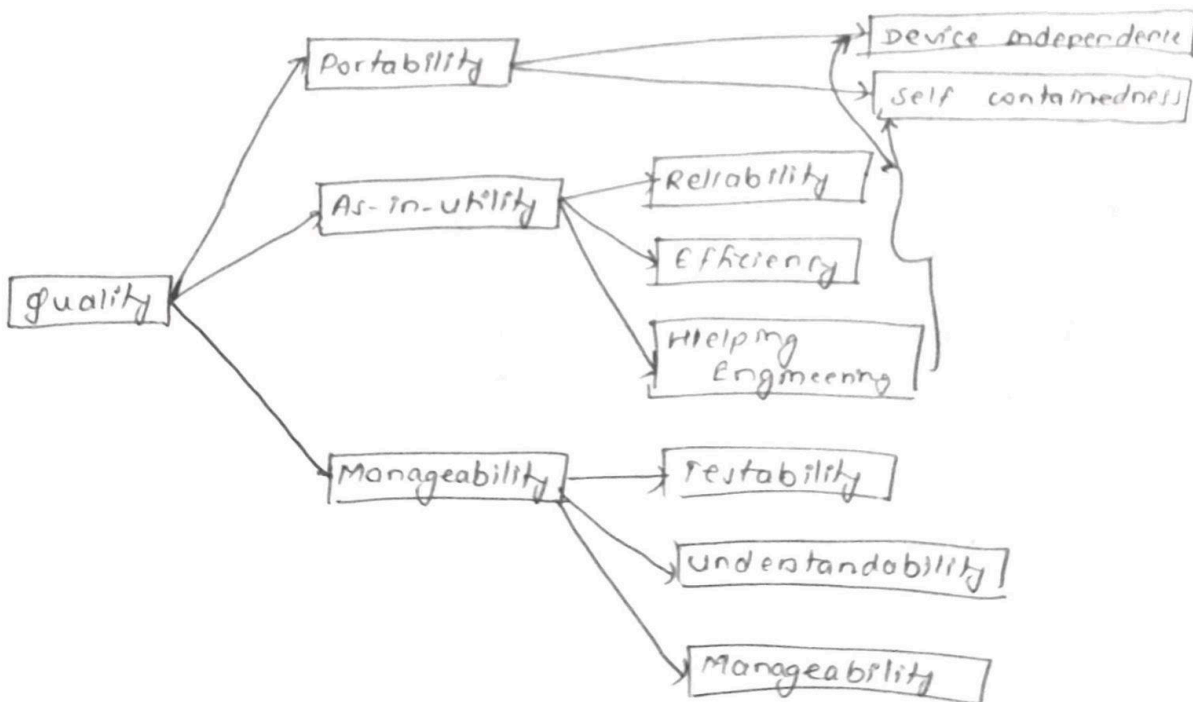


Fig. Boehm's software Quality Model

4.a)

ISO 9126 is an international standard.

It was developed in 1991

Provides a framework for defining and assessing quality of a software.

The six major external quality characteristics

1) Functionality - It specifies all the functions & tasks performed by the software

Sub characteristics

→ Usability - How much requirements of end users it satisfies

→ Interoperability - How it works with other systems

→ Security - Security features including penetration tests

→ Compliance - Adhering to the principles of ISO 9126.

2> Reliability : How the Ability of the System to maintain performance in various specified situations.

Sub characteristics

→ Maturity: Frequency of failure due to faults

→ Fault tolerance: Ability to operate despite faults

→ Recoverability: How easily can the system recover from faults.

3> Usability : The ease of using a system

Sub characteristics

→ Learnability - Ease of learning

→ Understandability - Ease of understanding

→ Operability - Ease to operate the system

→ Attractiveness → UI of the system.

4) Efficiency: It is the correlation between Performance & resources utilised

Sub characteristics

- Time behaviour
- Response time
- Resource utilisation.

5) Maintainability - Ease to maintain & modify the software

Sub characteristics

- Analyzability - Analyzing issues
- Modifiability - Changing the code
- Testability - Ease to test once modified

68 Portability: How easily can the software be transferred to different environments

Sub characteristics

- Installability
- co-existence
- Adaptability
- Self-contentness.

ISO 9001: 2000

It is an international standard introduced by British standard institutious

It addresses all software characteristics and provides guidelines to evaluate them.

Key requirements for QMS

14 Fundamental features

- customer focus - focus on all requirements of customer

- Leadership
- Involving People
- Process approach
- continuous Improvement

2> Accessibility

~~The software~~

Ensures proper definition & establishment of software development process

Focuses on quality rather than certification

3> Certification

Audit by accredited bodies to certify it adheres to ISO 9001:2000 guidelines

4> Principles

- customer focus
- leadership
- Involvement of people
- Process approach
- continuous improvement
- Factual predictions
- Mutual beneficial Relationships

5) Activities of QMS

- Identify user requirement
- Establish quality policy
- Design process
- Allocation of responsibility
- Resource allocation
- measurement & monitoring
- Analysis & improvement

6) Detailed Description

- Documentation - Documenting the process
- Management responsibility - overseeing if adheres to ISO 9001:2015
- Resource Management -

10/5

1) Capability Maturity Model (CMM) is developed by the Software Institute of Engineering (SIE) of the Carnegie University with the support of United States Department of Defense (US DoD).

The different levels of CMM are:-

1) Level 1: Initial

- The organization under this level work is the haphazard condition.
- ~~The~~ One team starts the project and then in between if the team member leaves the organization then a new team member is hired.
- It is very difficult to get all the requirements of the project to new member.
- So, the delivery of product on time here depends upon the members of the development team.

i) Level 2 Repeatable

- Organization within this level documents the plan and execute accordingly.
- Due to documentation, they can follow the same procedure to develop another software and there is a chance of positive repeatable.

ii) Level 3 Defined

- Organization within this level, define ~~only~~ not only the product but also the management requirement.
- The exclusive training is provided to the employee to achieve that defined requirements.

iii) Level 4 Managed

- Organization under level 4, manages all the process.
- Here, the track of every procedure is ~~is~~ maintained but not ~~is~~ implement to increase performance.

iv) Level 5 Optimizing

- Organization under this level, try to optimize the already existing plans.
- They try to develop the fast, cost effective software within a time constraints.

pros
 b)

The various techniques to help enhance software quality are:-

i) Increasing visibility

→ Here, the code of one programmer should be easily understood by another programmer.

→ Egoless programming.

ii) Procedure outcome

→ Every procedure are define in software development cycle and follow the same

iii) check intermediate case :

→ Earlier, the whole ~~to~~ working model was developed and then debugged it to get the desired quality.

→ Now, the small part of the software, i.e. one component is developed and checked instantly and correct the errors.

The other techniques to enhance software quality are:-

iv) Inspection: The ~~product~~ component of the software is developed and the copy of it is shared among colleagues or members to check the quality.

v) Formal method: Before, a mathematician team was there to check the ambiguity and unambiguity of the code by implement different logics.

now, the object functional language (OFL) is used so that it removes ~~the~~ ambiguity and help for non-mathematical developer to check the quality.

iii) Software Quality Circle (SWQC)

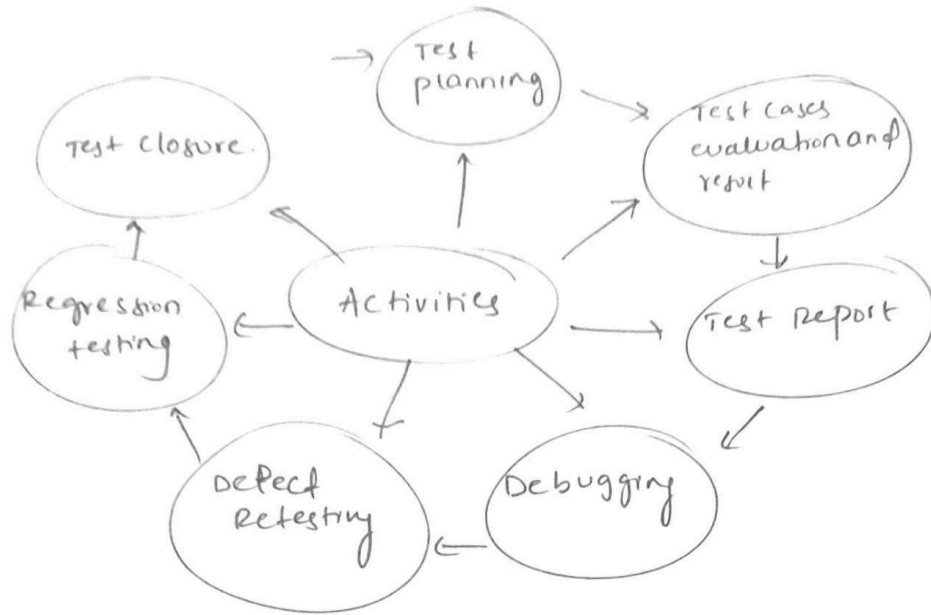
- It is a Japanese quality circle.
- Here, a quality team is formed in each department to check quality.
- The source of error is found and then it is used not to repeat the error again.

iv) Clean room

- Here the different departments are separated
- Specification team, development team and ~~testing~~ certification team.

6.
(1)

Major activities involved in software testing.



① Test planning in this activity all the preplanning to a test process is defined.

→ the simulations, softwares or test bed is defined.

② Test cases evaluation and result verification.

→ Here the testcases are applied for software and are evaluated on basis of expected and actual results.

→ The Actual results are noted or documented.

③ Test Report

- The errors / failed test cases are formally documented.
- They are sent for developers for debugging phase.

④ Debugging

- The developers analyze the test report and then they will understand and spot errors.
- They rectify the mentioned errors in the report.

⑤ Defect Retest

- Post debugging and error validation the testing team retests the code for errors found out previously.
- If all noted errors are cleared it is proceeded for next phase.

⑥ Regression testing

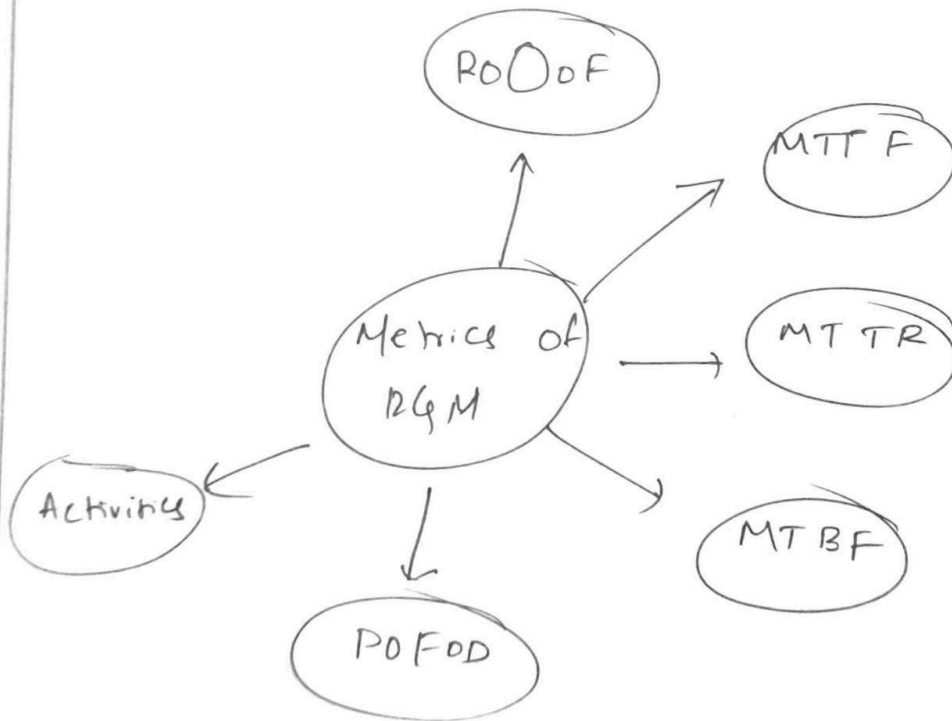
- The unmodified features or codes are retested here.
- Because the debugging phase might create new errors to unmodified parts of code.
- If all cases are passed then the code is sent to ^{to} further.

Test Closure

→ The testing code is ended with all logged and documented data for future references.

6
(b)

Six Reliability metrics of Reliability Growth model



① Rate of occurrence of failures:

here the rate of occurrence of failure is

measured.

→ The mean ^{Rate} of all no. of failures occurred over a period of time is calculated.

② MTTF (Mean time to failure)

→ It calculates the mean time between the failures occurred over a period of time.

→ given by formula $\frac{\sum t_i}{n}$ where $\sum t_i$ is time of each failure and n is no. of failure.

③ Mean time to Repair (MTTR)

→ Once the failure is identified it is sent to repair and later added to main code.

→ The mean time between all repairs from identification to clearance and re-deploy into code is called MTTR.

④ MTBF (Mean time between Failures)

→ It is the time between the failures and is calculated as sum of MTTF and MTTR.

$$\boxed{MTBF = MTTF + MTTA}$$

⑤ Probability of Failure on Demand (POFOD)

→ It is the probability of failure occurred when the software is requested or demanded.

→ Ex. if error / failure occurs at 1 time in 1000 requests
Probability = $\frac{1}{1000} = \underline{0.001}$

⑥ Activities

→ The activities involved in software also contribute for reliability metrics

→ Different activities acts differently over a software

Different types of RQM's are as follows.

① Goel - otulcomo model

② Littlewood - verraf model

③ Jelinski - morando model