

# Software engineering

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

## 1. Introduction to System Concepts.

### Q.1 Explain and define System?

- ① System is defined as "an integrated collection of related Components that interact to perform a task in order to accomplish a goal".
- ② The word 'System' is derived from Greek word 'Systema' which means to 'place together or to combine'.
- ③ System is an orderly grouping of interdependent Components linked together acc<sup>n</sup> to plan to achieve specific goals.

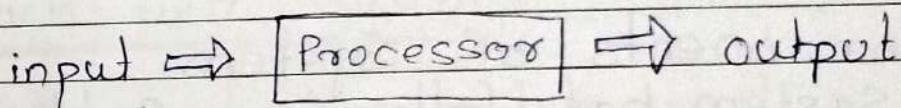
### Q.2 What are the characteristics of System?

- ① Organisation - It implies structure & order.
- ② Interaction - it defines manner.
- ③ Interdependence - it means how Components of system depend on one another.
- ④ Integration - it means how System is tied together in order to achieve goal.
- ⑤ Central Objective - it means common goal and it may be real or stated. The objective of System must be central.
- ⑥ Behaviour - it is the way how System reacts to its surrounding environment.
- ⑦ Structure - it defines boundary bet<sup>n</sup> system and environment.

Q.3 Explain Components of System? OR  
Explain Elements of System.

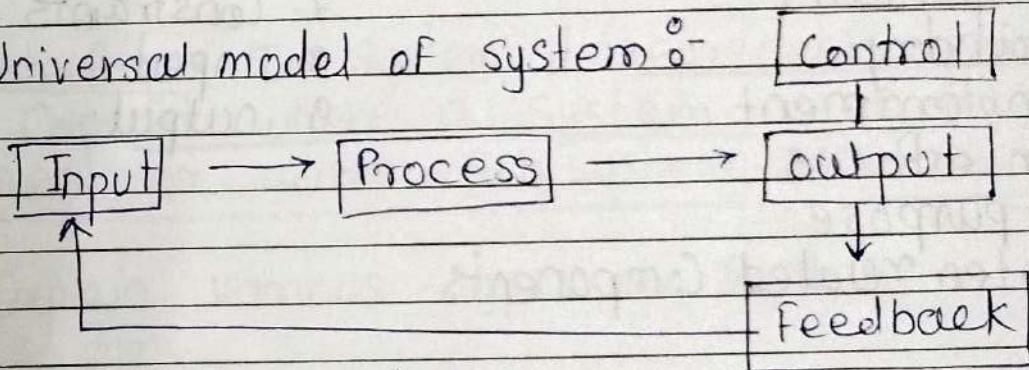
- ① The basic System Components or elements are Input, processor and output.
- ② Universal model of System is made up of System elements like input, processor and output using basic Concept like Control and Feedback to keep System balance.

③



General model of System.

④ Universal model of system :-



⑤ Inputs and outputs :-

① inputs are the information that enters into System for processing

② output is the outcome of processing.

⑥ Processor :-

① it is Operational Component of System

② it involves actual transformation of input into output.

### ⑦ Control :-

- ① it controls working of system at all stages
- ② it controls all input, processor and output activities

### ⑧ Feedback :-

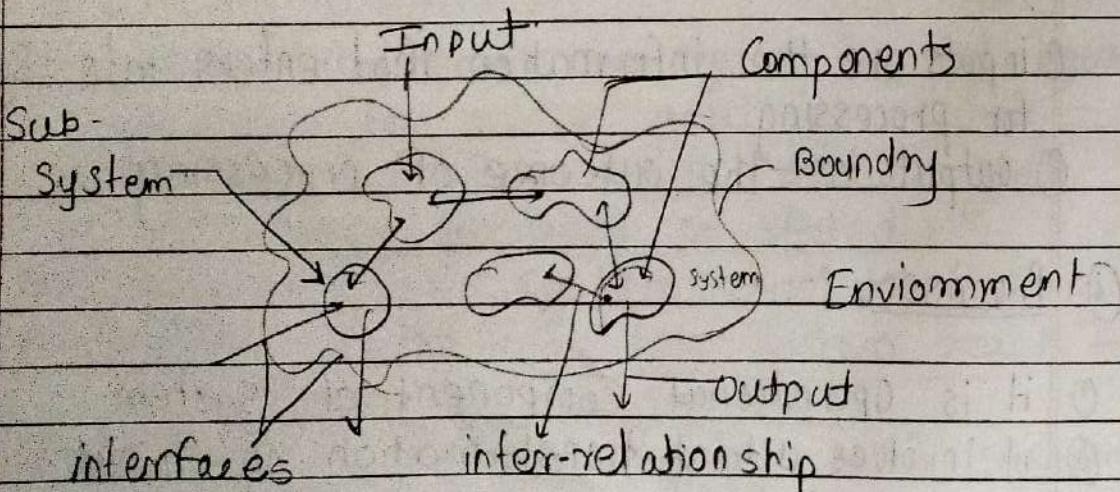
- ① Feedback is the method that helps to compare output produced with output expected.

### Q.4. Components of System.

→ ① \* System has following System Components :-

- |                              |                |
|------------------------------|----------------|
| 1. Components                | 7. Constraints |
| 2. Boundary                  | 8. input       |
| 3. Environment               | 9. output      |
| 4. Interfaces                |                |
| 5. A purpose                 |                |
| 6. Inter-related Components. |                |

### ⑨ Diagram.



1. Sub-system/Components - A Sub-System is either a complex part or an aggregate of part called as sub-System;
  2. Boundary - Boundary of a System is the line that marks the inside and outside of a System;
  3. Environment - Environment in a System represent everything external to a System that interacts with System;
  4. Constraint - it is a limit to what a System can accomplish.
  5. Interface - In system it represents point of contact where a System meet its environment or where sub-system meet each other.
- Q.5 Explain various types of System in detail.

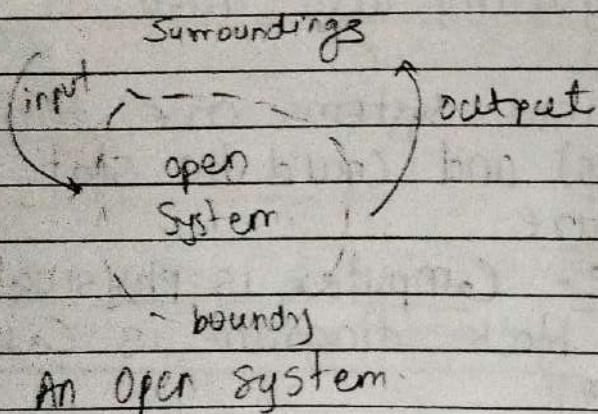
- 
- I] Conceptual (abstract) and physical System :-
- Abstract Systems are represented conceptually not physically also called as non-physical System;
  - The physical Systems are tangible (Countable entities) and could be static or dynamic in nature
  - Example - Computer is physical System and its block diagram is called abstract System

## II] Deterministic and Probabilistic Systems.

- ① Deterministic System works in predictable manner.
  - In deterministic System occurrence of all events is perfectly predictable.
  - example - in Computer outputs are deterministic.
- ② A probabilistic System is one in which occurrence of events cannot be perfectly predictable or predicted;
  - example - weather forecasting system is probabilistic;

## III] Open and close Systems.

- ① Open System.
- ② In Open System, System Continously interacts with its environment
- ③ An open System is one which does not provide for its own control and modification.
- ④ it does not Supervise itself, if needs to be Supervised by people.



example-

## ① Close System

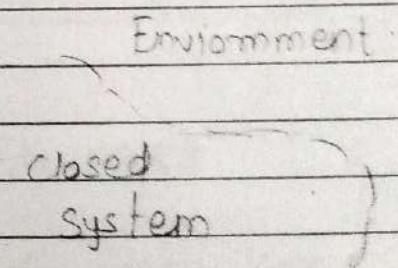
- ⑤ open system have input and output flow. it represents exchange of matter energy with their Surrounding.
- ⑥ example - in education System, Students info shared with companies for placement
- ⑦ Open System have many interfaces with environment.

## ② Closed System -

- ⑧ A System in which does not interact with the outside environment is known as the closed System.
- ⑨ Closed System has no input or output.
- ⑩ A closed system is one which automatically controls or modifies its own operation by responding to data generated by system itself.
- ⑪ A closed System is a System in the state of being isolated form the environment.

⑫

diagram -



boundary-  
Closed System

- ⑬ example - A Computer System.

#### IV] Natural and artificial System.

- Natural system exist and also bound in nature like rivers, mountains.
  - Artificial systems are manufactured (man-made) such as dams, rivers, canals, roads etc.
  - Example - Solar System.
- \* Long forms.

- ① CBIS - Computer based information System
- ② DSS - A Decision Support System
- ③ TPS - Transaction processing System
- ④ MIS - management information System
- ⑤ OAS - office automation System
- ⑥ ES - expert System
- ⑦ EIS - executive Information System.  
Reed definition from textbook at exam time.

## 2. Introduction to Software Engineering

Q.1 What is Software?

→ Software is more than just a program code.

Q.2 What is engineering?

→ Engineering is all about developing product using well-defined, scientific principles and methods.

Q.3 What is Software engineering?

→ ① Software engineering is defined as "the application of systematic, disciplined, quantifiable approach to development of operations and maintenance of software".

② Software engineering is concerned with development and maintenance of technological product, problem solving techniques etc.

③ A primary goal for Software engineering is to improve the quality of Software products.

Q.1 What is Software?

→ ① It is a set of instructions that when executed provide desired output, performance and functions.

Q. 4 What are characteristics of Software?

- ① Software is developed or engineered Software do is not manufactured in the classical Sense;
- ② Software does not 'wear out'
- ③ Most Software is Custom Built, rather than being assembled from existing Components.

Q. 5 Explain Software application domains.

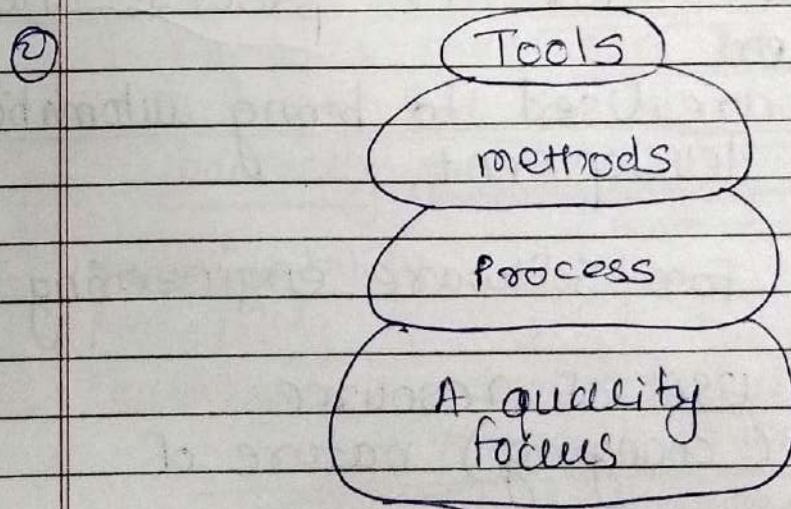
\* There are Seven main Categories of Software domains.

1. System Software - it is collection of program written to services other programs. example - Compilers
2. Application Software - it is stand-alone program. example - Payroll System.
3. Engineering / scientific Software - it uses various number crunching algorithm. example - System Simulation.
4. Embedded Software - it resides within a Software or product; it can perform limited and esoteric function; eg- software in mobile phone.

5. Product line Software - it is developed / designed to provide specific capability for used by different customers.  
eg - spreadsheets, multimedia, DBMS etc.
6. Web applications - also called as "web apps".  
eg - online forms, word processors etc.
7. Artificial Intelligence (AI) software - it uses various non-numerical algorithm to solve complex algorithm.  
eg - robotics, expert systems.

Q.6 Explain layered Technology of Software engineering.

- ① layer technology of Software engineering consists of set of key elements i.e quality focus, process, methods and Tools etc.



layers of layered technology of Software engineering.

## I. A quality Process layer:-

- ① an engineering approach must have a focus on quality.
- ② The software product should fulfill the customer quality requirements.

## II. Process layer :-

- ① Foundation for software engineering is the process layer;
- ② Software engineering process holds all technology layers together.

## III. Methods layer :-

- ① it provides technical knowledge for developing Software,
- ② The method layer covers a broad array of tasks.

## IV. Tools layer :-

- ① it provides automated or semi-automated support for the process and method layers.
- ② These tools are used to bring automation in software development.

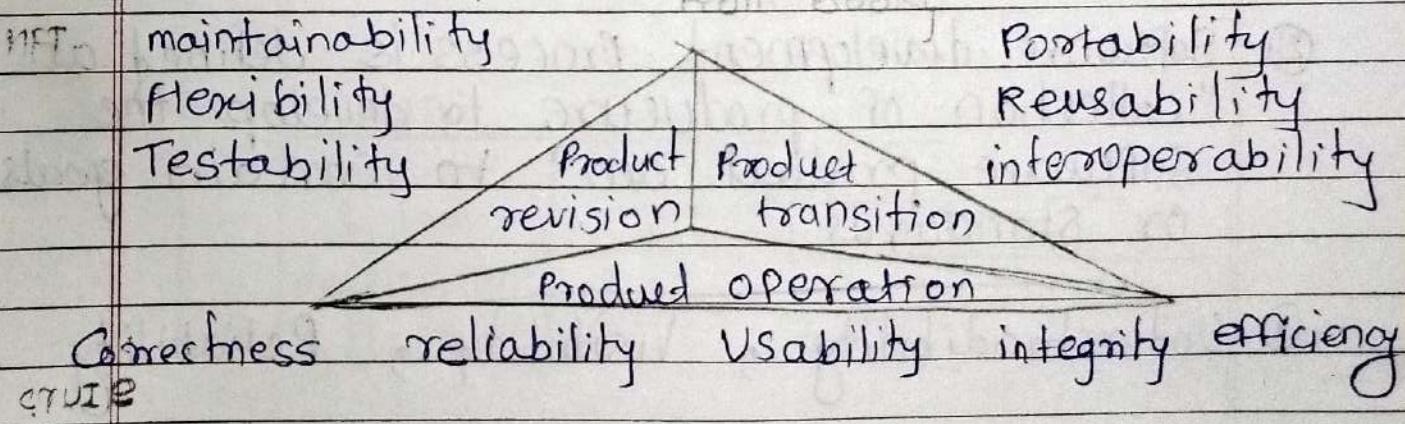
## Q.7 Explain need for Software engineering.

- ① for efficient use of resources
- ② for dynamic (changing) nature of Software.
- ③ for reduction of Complexity.

- ④ For quality management
- ⑤ For reduction of cost and time
- ⑥ For Production Reliable Software
- ⑦ To maintain effectiveness of Software
- ⑧ For production large Software
- ⑨ For improve scalability
- ⑩ To increase productivity

### Q.8 Explain Mc Call's Quality Factors

- ① Quality factor refers to the characteristics of a product or services which defines its ability to satisfy user requirement.
  - ② IEEE defines Software quality as "the degree to which a System Components or process meets Specified requirements".
  - ③ There are several factors which are used as measure to enhance the quality of Software.
  - ④ The factors that affect Software quality can be categorised in two broad grp's.
    - i. factors that can be directly measured
    - ii. factors that can be measured only indirectly
- (detected at the point of first failure)  
Explanation needs  
From Book)



Mc call's Software Quality Triangle

## ⑤ Product operation

- i. it checks whether the software is able to perform the desired function efficiently
- ii. it includes :- correctness, Reliability, Usability, integrity, efficiency.

## ⑥ Product Revision

- i. it checks whether the software requires efforts to modify.
- ii. it includes maintainability, flexibility, Testability.

## ⑦ Product transition

- i. it checks whether the software is adaptable to the new environment
- ii. it includes following factors - Portability, Reusability, Interoperability

## Q.7 What is Software process? Define.

→

① Software Process is defined as "the related set of activities and process that are involved in developing and evolving a Software System."

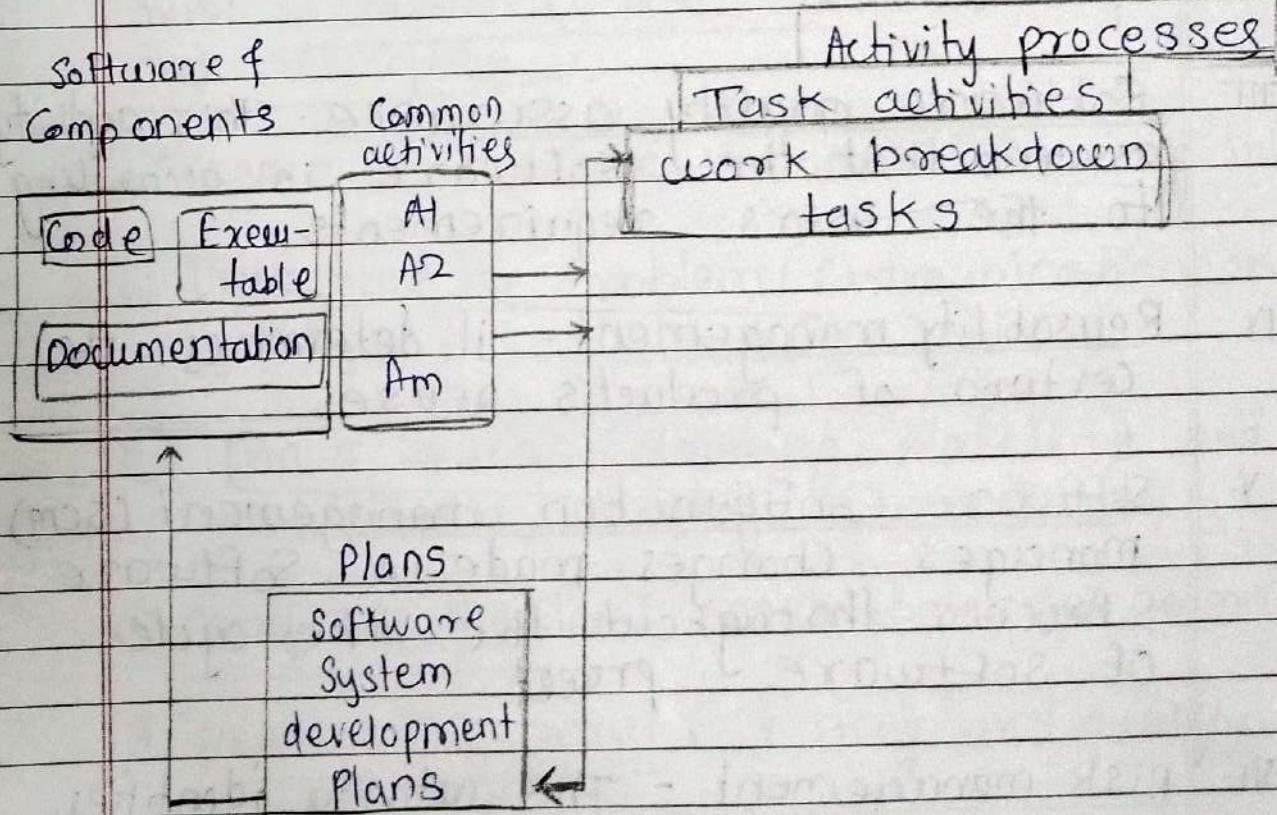
② Software development process is defined as "collection of procedure to develop the Software product acc" to certain goals or standards"

③ Understandability, Visibility, Reliability

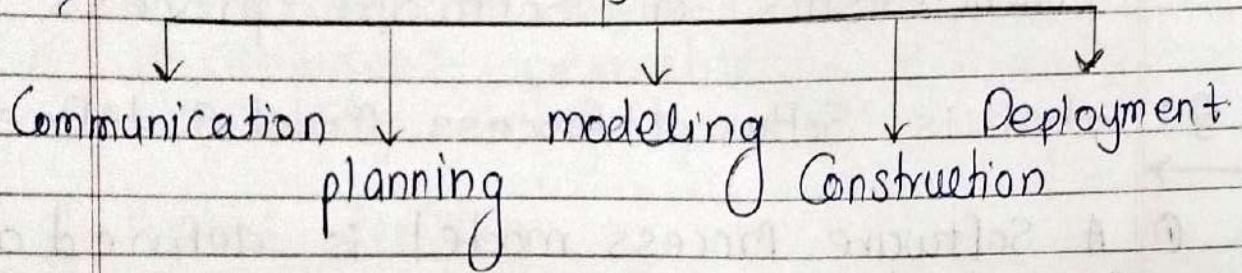
Supportability, Rapidity, Acceptability, Robustness, maintainability are the some characteristics of Software process

Q.8 what is Software Process Model ? define.

- 
- ① A Software Process model is defined as "a strategy ~~Comprising~~ process, method and tools layers as well as general phases for developing the Software".
  - ② A process model for software engineering depends on the nature and application of the software project.
  - ③ Basic Generic Software Process model



Q.9 Software Process framework Activities includes following activities.



Q.10 Explain Umbrella activities.

- I Software Project tracking and control - it tracks the software development process and controls daily work etc.
- II Formal Technical Reviews (FTR) - This activity Consider the Code, products and documents of software engineering practices to detect and remove errors.
- III Software quality assurance - This activity assures that the software is according to the users requirements.
- IV Reusability management - it determines the criteria of product's reuse.
- V Software Configuration management (scm) manages changes made in Software Process throughout the life cycle of software project.
- VI Risk management - This activity identifies, analyses, evaluates and eliminates the

## Possible results

### Q.11. Software engineering practice.

① Software engineering practice is a collection of concepts, principles, methods, and tools that a software engineer calls upon on a daily basis.

\* The essence of practice.

② Essence is a standard that defines the smallest set of concepts that are common to all software project.

③ Practice is a broad array of concepts, principles, methods and tools that you must consider as software is planned and developed.

④ Essence of Practice includes following things

① Understand the Problem (communication and analysis)

② Plan a Solution (planning, modeling and software design)

③ Carry out the Plan (construction, code generation)

④ Examine the result (testing and quality assurance).

Q.12 Explain General Principles proposed by David Hooker.

- 
- + David hooker has proposed Seven general (core) principles which focus on Software engineering practices as whole.

Principle 1 :- Remember the reason that the Software exists.

- A Software System exists for one reason to provide value to its users.

Principle 2 :- keep it Simple

- design should be Simple so that System must be easily Understand and easily maintained.

Principle 3 :- Maintain the vision of the project

- A clear vision is essential for the Success of Software project.

Principle 4 :- what you produce, other will Consume

- Always design; Specify and Implement knowing that Someone else will later have to Understand and modify what you did.

Principle 5 :- Be open the future

- ✓ A System which has long lifetime has more Value.

Principle 6 :- Plan ahead for Software Reuse.

- ✓ Reusability saves time and effort.

Principle 7 :- Think then act.

- ✓ Thinking again the knowledge about how to do things right.

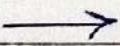
\* Defination of Umbrella activities.

→ In addition to framework activities, process, framework also Comprises a set of activities known as Umbrella activities.

It is used in Software Process.

### 3. Software development life Cycle (SDLC) and methodologies.

Q.1 What is SDLS ? Explain.



- ① A software life cycle is a series of identifiable stages that a software product undergoes during its life cycle.
- ② Software development life cycle was developed by National Computing Centre in 1960.
- ③ SDLC is defined as "a process or series of steps or phases that provide a model for the development of high quality software".
- ④ SDLC makes the development team able to design, create and deliver a high-quality product.
- ⑤ SDLC concentrates on feasibility analysis, cost benefits analysis, project management etc.
- ⑥ The process used to create a software product from scratch up to its public release is called SDLS SDLC.
- ⑦ The period of time during which these activities takes place are called phases.

⑧ The SDLC is a set of steps which are used for building a System;

⑨ Steps or phases of SDLC :-

PJ&FS, A, D, I, T, D, M

I] Primary

investigation and feasibility study

→ ② Analysis

→ ③ Design

→ ④ Implementation

→ ⑤ Testing

→ ⑥ Deployment

→ ⑦ Maintenance

II] Primary investigation and feasibility study :-

- Primary investigation and feasibility study is the first phase of SDLC.
- Primary investigation also known as Preliminary system study.
- During Primary investigation phase all relevant information is collected from the customer to develop a product as per their expectation.
- Feasibility study is basically the test of the proposed system.
- During, Feasibility study, User determines the overall Project Scope.

II] Analysis :-

- System analysis is the process of studying

a procedure ~~or business~~ in order to identify its goals.

### III] Design :-

- System design refers to the process of planning for a new system.
- In this phase, the requirement gathered in the SRS document is used as an input.

### IV] Implementation of Coding :-

- Implementation starts once the developer gets the Design document.
- The Software design is translated into Source code.
- All the Components of the Software are implemented in this phase.

### V] Testing :-

- Testing phase starts once the Coding is Complete.
- In this phase, developed Software is tested and if error comes developer fix it.

### VI] Deployment :-

- Once the product is tested it is deployed in the production environment.

- In this phase, first UAT (User acceptance testing) is done depending on the Customer expectations.

## VII] Maintenance :-

- After Software is deployed, its maintenance begins.
- This is the last phase of SDLC.
- The Procedure where the Care is taken for the developed product is known as maintenance.

## ⑩ Advantages of SDLC :-

- A Common Vocabulary for each step.
- defined communication channels between development team and stakeholders.
- clearly-defined inputs and outputs from one step to the next.

## \* Types of SDLC Generic Process model

### ① Descriptive model

### ② Prescriptive model

a. waterfall  
model

b. Incremental evolution  
model

c. V and V model

a. prototypic  
model

b. Spiral  
model

Q. What is SDLC Generic Process models

\* Generic Process Model :- CPMCD

- ① A Software process is a set of activities and associated results, which produce a Software Product.
- ② A Software Process model is an abstract representation of a Software Process;
- ③ A Process model is defined as "a model of a Process System that describes Process Organization, Categorization, hierarchy, interrelationship."
- ④ Software Process models are systematic methods for controlling and coordinating the development of Software product achieving all stated objectives.
- ⑤ Generic Process model includes following 5 (five) main activities :
  1. Communication - involves Communication betn stakeholders and customer
  2. Planning - Establishes plan for Software development.
  3. Modelling - it includes Analyze, design creation of models to better.
  4. Construction - it includes Code and test it combines code generation and testing to find the errors.

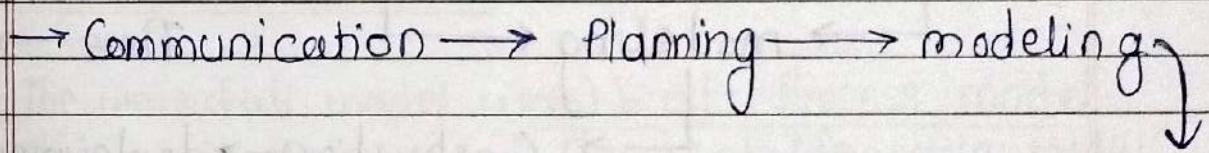
- 5. deployment - Involves delivery of software to the customer.

#### (6) Advantages of Process Model :-

1. Enables Effective Communication - it enhances Understanding.
2. Facilitates Process Reuse - it utilizes the existing process for different projects.
3. Effective - process model can be used again and again.
4. Facilitates Process management.

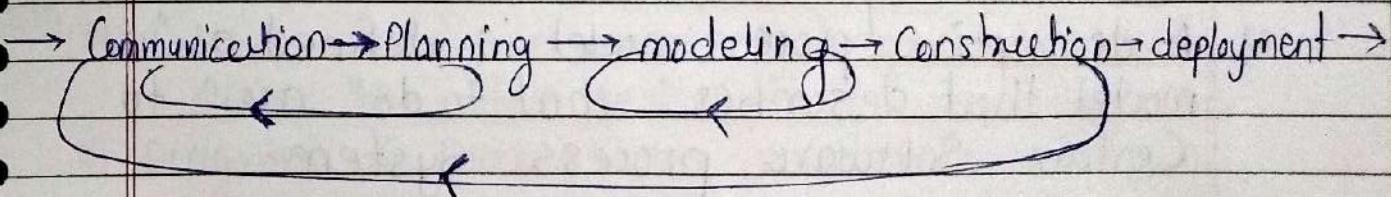
#### \* Types of Process Flows in SDLC :-

① Linear Process Flow :- This flow executes each five activities in sequence.

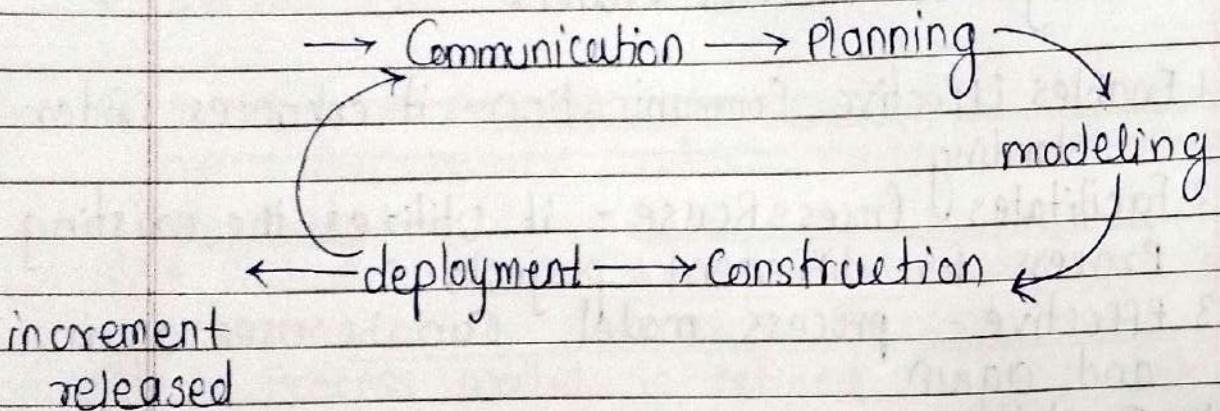


Construction  
↓  
← deployment ←

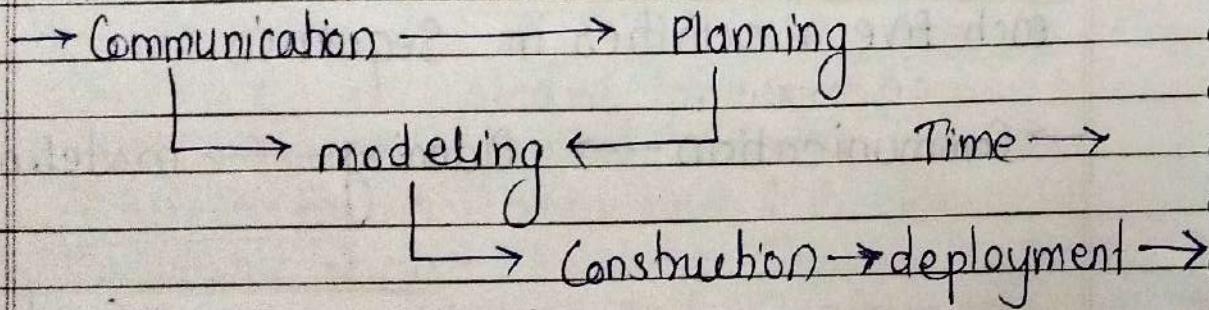
② Iterative Process Flow :- This flow repeats one or more activities before proceeding to the next.



- ③ Evolutionary Process flow - This flow executes activities in a 'Circular' manner.



- ④ Parallel Process flow:- This flow executes one or more activities in parallel with each other activities.

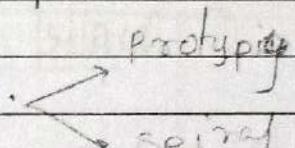


### Types of SDLC Process model :-

#### I] Descriptive model:-

- 1) A descriptive model describes the history of how a particular software System was developed.
- 2) A descriptive process model is defined as "a model that describes 'what to do' all^n to Certain Software process System."

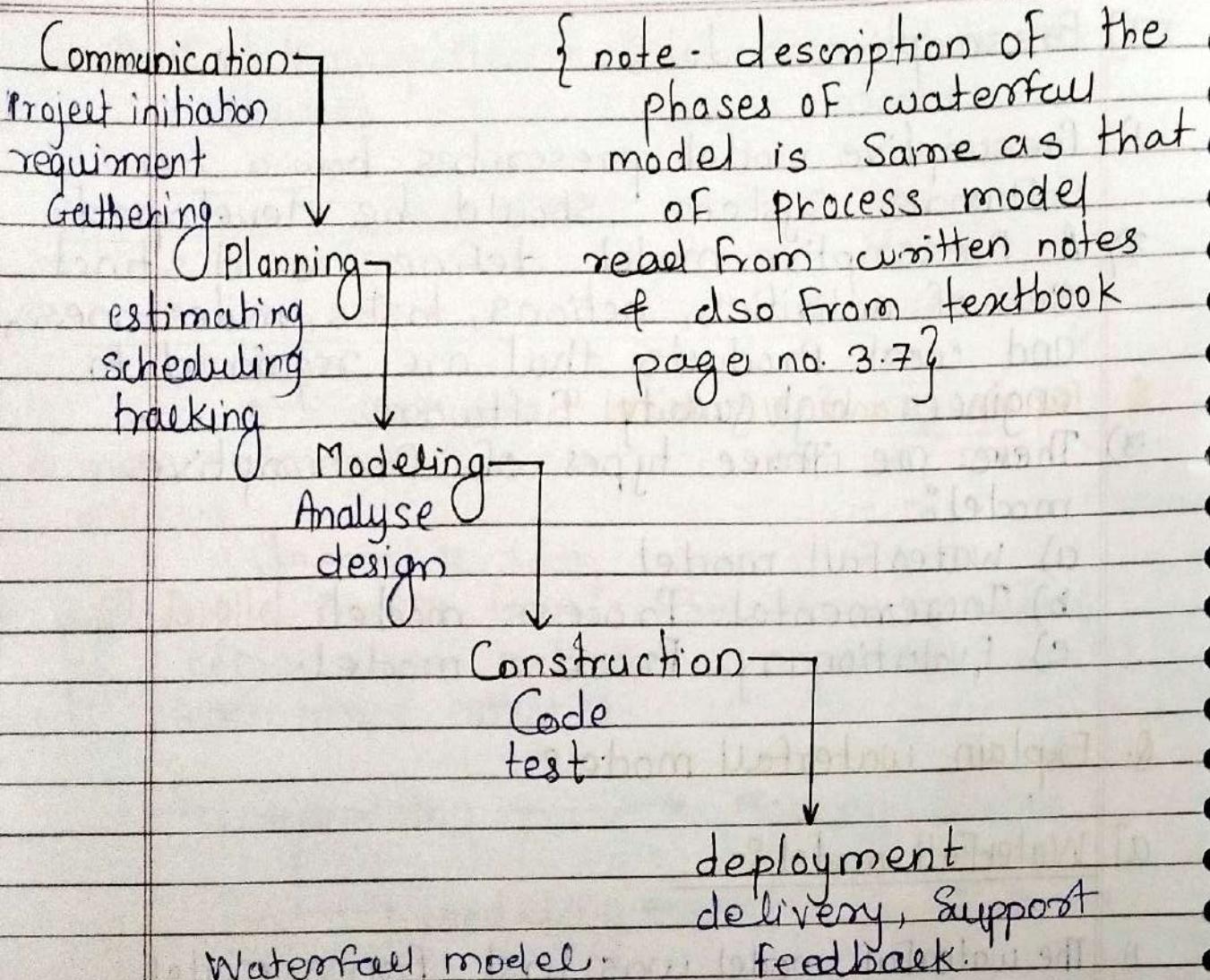
## II] Prescriptive model :-

- 1) Prescriptive model prescribes how a new Software System Should be developed;
- 2) A Prescriptive model defines a distinct Set of activities, actions, tasks, milestones, and work products that are required to engineer high-quality Software.
- 3) There are Three types of prescriptive model :-
  - a) waterfall model and V model
  - b) Incremental Process model
  - c) Evolutionary Process model. 

Q. Explain waterfall model?

### a) Waterfall model :-

- 1) The waterfall model was First Process model
- 2) It is also known as "Classical life cycle model"
- 3) It is traditional software lifecycle model.
- 4) In waterfall model, a work flow is in linear (sequential) fashion or Pattern.
- 5) The waterfall lifecycle model is defined as "the model of building in stages, whereby one stage is completed before next stage begins".
- 6) In waterfall model, each phase must be completed fully before the next phase can begin.
- 7) In this model Software testing starts Only after the development is complete.
- 8) Diagram of waterfall model :-



### g) Advantages:-

- This model is very easy, simple to understand and to use.
- This models are basically used for small projects.
- In waterfall model phases do not overlap.
- In this model progress of system is measurable;
- The amount of resources is minimal.

## 10) Disadvantages:-

- a. It is poor model for long and ongoing projects.
- b. It has high amount of risks and uncertainty.
- c. it does not support incremental iteration, so changes can cause confusion.

Q. Explain V and V model :-

a.i) V and V model :- (Diagram is remaining)

- 1) V and V model is an extension of waterfall model.
- 2) V and V model stands for Verification and Validation model.
- 3) In this model, each phase of SDLC must complete before the next phase starts.
- 4) V and V model also follows sequential design process same as waterfall model.
- 5) Testing of the device is planned in parallel with a corresponding stage of development.
- 6) The phases of testing are "Validation phase".
- 7) The phases of development as "Verification phase".
- 8) <sup>each</sup> Phase of development there is corresponding test activity planned in advance.
- 9) Phases of V and V model :-

➤ Phases of Verification Phase of V and V model -

- I. Business requirement analysis:- This is first step where product requirements understood from

development.

## the Customer Side

- II. System design - In this phase, design of system hardware and Software Specification is made.
- III. Architectural design - This phase is concerned with drafting and technical methodologies.
- IV. Module design - This phase, the System breaks down into Small modules.
- V. Coding phase - after designing Coding phase is started, Suitable programing language is decided.

## Phases of Validation phase of V-model :-

- I. Unit testing - Unit Test Plans (UTPs) developed during the module design.  
Unit testing verifies that the Smallest entity can function correctly.
- II. Integration testing - integration testing plans are developed during the architectural design phase.
- III. System testing - System testing plans are developed during system design phase.  
System testing is performed when the Complete System is ready.
- IV. Acceptance Testing - it is related to business requirement analysis part.  
it includes the testing Software product in user atmosphere.

## 10) Advantages :-

- a. Easy to Understand.
- b. Works well for Small projects.
- c. This model avoid downward flow of the defect.
- d. This model ~~saves~~ saves a lot of time.
- e. higher chances of success.

## 11) disadvantages :-

- a. Very rigid and least Flexible.
- b. Not a good for Complex project.
- c. No early prototypes of software are produced.

## Q. Explain Incremental Process Model. 9

### b) Incremental Process model :-

- 1) Incremental model is defined as 'a model of software development where the product is designed, implemented and tested incrementally until the product is finished.'
- 2) In incremental model, multiple development life cycle take place.
- 3) In this model, cycles are divided up into smaller modules.
- 4) The work flow is in a linear (sequential) fashion within an increment.
- 5) In incremental model, multiple independent deliveries are identified.
- 6) Incremental model provides a needed set of functionality.

## 7) Diagram :-

Increment #1

Communication

Planning → modeling → construction → deployment

delivery of 1<sup>st</sup> increment

Increment #2

Communication → planning → modeling → Construction → deployment  
delivery of 2<sup>nd</sup> increment

Increment #n

Communication → planning → modeling → Construction → deployment  
delivery of n<sup>th</sup> increment

## 8) Advantages :-

- a. It is Used when requirements are well understood.
- b. It model is Useful when staffing is too short.
- c. Lowers initial delivery cost.
- d. In this model, Customer Can respond to each built.
- e. it is easier to test and debug.
- f. This model is flexible to change scope.
- g. Easier to manage risk;

## 9) Disadvantages :-

- a. Needs Good planning and design.
- b. Total cost is higher than waterfall model.
- c. Needs a clear and complete definition of whole system.

### c) Evolutionary Process Model :-

- ① Evolutionary model are iterative.
- ② Evolutionary model enables to you to develop increasingly more complex version of software.
- ③ Evolutionary model used for complex projects.
- ④ Evolutionary model has basic two models i.e.
  - 1. Prototyping and 2. Spiral Model.

### Q. Explain Prototyping model ?

#### 1. Prototyping model :-

- 1) Prototyping follows an evolutionary and iterative approach.
- 2) Prototyping model serves to clarify requirements, which are not clear.
- 3) Prototyping model is used when requirements are not well understood.
- 4) Prototyping model focuses on those aspects of the Software that are visible to customer.
- 5) Phases of Prototyping model :-

I. Requirement Gathering - in this phase, product requirements are gathered. developer and customer meet and discuss objectives of software.

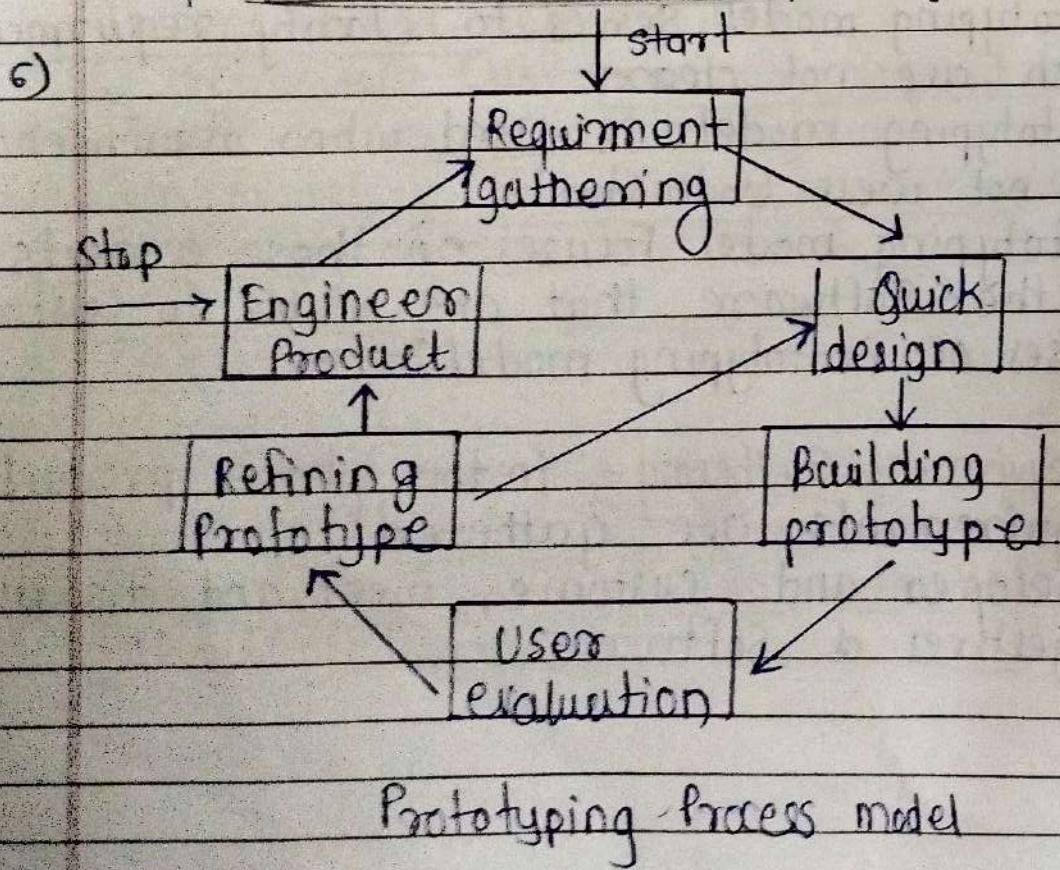
**II. Quick design-** Quick design is created when requirements are known. It focuses on representation of software that is visible to customer.

**III. Building prototype-** Information gathered in quick design leads to the construction of prototype.

**IV. User evaluation-** Evaluation of prototype recognizes its strengths and weakness. i.e. what is to be added or removed.

**V. Refining prototype-** In this phase, if user is not satisfied with current prototype then it improves according to requirements.

**VI. Engineer Product-** In this phase, user accept the end prototype when all the requirements are completely met.



## 7) Types of Prototyping Model :-

- I] Rapid Throwaway Prototyping - In this prototype is developed rapidly.
- II] Evolutionary Prototyping - In this prototype is made and the client feedback is received.
- III] Incremental Prototyping - In this prototype model, final product requirements are broken into smaller parts.
- IV] Extreme Prototyping - This Prototype method, used for web applications.

## 8) Advantages :-

- a) Errors are detected much earlier.
- b) Gives quick user feedback for better solutions.
- c) Missing functionality can be easily identified.
- d) Development process of this model users are actively involved.

## 9) Disadvantages :-

- a) It is slow process.
- b) it takes more time to develop.
- c) changes can distract the rhythm of development team.
- d) Client involvement is more.

8 Explain Spiral model ?

## 2. Spiral Model :-

- 1) Spiral model is defined as "a model of the software development process in which the constituent activities like typical requirements analysis, design, coding, testing performed iteratively until the software is complete".
- 2) Spiral model was invented by Dr. Barry Boehm in 1988.
- 3) Spiral model follows evolutionary approach.
- 4) Spiral model is combination of waterfall model and interactive model.
- 5) Each phase in Spiral model begins with a design goal and ends with client reviewing.
- 6) Inner Spirals focus on identifying software requirements and project risks.  
Outer spirals take on classical waterfall approach.
- 7) Risks analysis is the most important criteria considered in this model.
- 8) The Spiral model has five phases Communication, Planning, Risks analysis, modeling, construction, and deployment.
- 9) Each subsequent Spiral builds on the baseline Spiral.
- (10) Diagram :-

## Spiral model

Communication

Planning  
(estimation,  
Scheduling,  
Risks  
analysis)

deployment  
(delivery,  
feedback)

modeling  
(analysis,  
design)

Construction  
(Code, test)

### ii) Advantages :-

1. It focuses on early error detection and design flaws.
2. it incorporates prototyping as a risk-reduction strategy.
3. it incorporates Software quality objectives into the product.

### 12) Disadvantages :-

1. Spiral model can be a Costly model to use.
2. it is not suitable for low risks Projects.
3. High chances of risk analysis.

## Q. What is Concurrent Models?

- ① The Concurrent process model shows the current state of activities, tasks and their associated state that remain in different phases.
- ② The Concurrent model defines a series of events that will trigger transitions from state to state for each of software engineering activities, actions and tasks.
- ③ advantages -

- It provides the accurate picture of current state of Software model.
- This modeling is applicable for all types of Software development.
- This model gives immediate feedback from testing.
- It is flexible.

## ④ Disadvantages -

- Analysis and designs are complex tasks.
- It focuses on mainly flexibility and not on quality.

# 4. Requirement Engineering

Q.1. What is Requirement? what are types of requirement?

- ① Requirement is defined as "a Condition or Capability needed by a User to solve a problem or achieve an Objective / goal".
- ② Requirement is Feature of System.
- ③ Types of requirement:

## 1. Functional Requirement -

- Functional requirement is defined as "a function that a system must be able to perform".
- These requirement describe the interaction of software with its environment.

## 2. Non-functional Requirement -

- The non-functional requirements also known as quality requirement.
- These requirement is related to system attributes, such as reliability and response time.

## 3. Domain Requirement -

- The requirements that are derived from the application domain of system, instead from the needs of the users are known as Domain requirement.

at  
gr  
an  
to  
principles

Q.2 what is requirements Engineering?

- 
- ① Requirements engineering is defined as "the systematic process of documenting requirements through an interactive co-operative process of analyzing the problem, documenting the resulting observations in a variety of representation format, and checking the accuracy of understanding gain."
  - ② Requirements engineering is a description of what the system will do without describing how it will do.

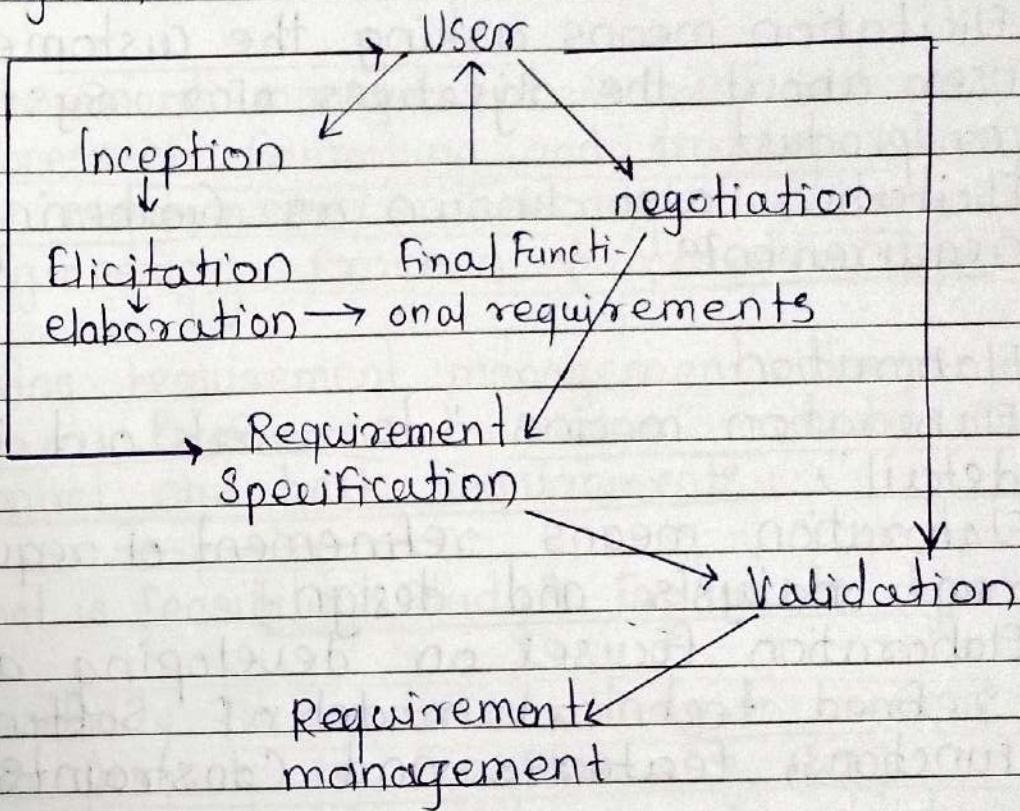
Q.3 What is the need of the requirement engineering?

- 
- Requirement engineering is an important stage in any software development.
  - Requirement engineering helps achieve primary objectives of making sure that the delivered system meets the customer needs.
  - Requirement engineering provides mechanism of understanding what the customer, user wants.

Q.4 What is Requirement engineering Process?

- 
- ① The process to determine the requirement specification of the software is called the requirement engineering process.

## ② Diagram



Q.5 Explain in detail the tasks of the requirement engineering.

- There are Seven distinct tasks in requirement engineering
- All these 7 tasks collectively form the strong base for software design and constructions;
- tasks as follows:- IEENSVR

### I. Inception -

- inception means beginning.
- inception tells about how does Software Project gets initialized;
- it establish basic understanding of problem and nature of solution.

## II. Elicitation

- Elicitation means asking the customer, user about the objectives of system or product
- Elicitation also known as Gathering of requirements,

## III. Elaboration

- Elaboration means "to work out in detail."
- Elaboration means refinement of requirement analysis and design
- Elaboration focuses on developing a refined technical model of software functions, features and constraints

## IV. Negotiation

- Negotiation is a discussion on financial and other commercial issues.

## V. Specification

- Specification is the final workproduct
- It means different things to different people,
- It describes the function and performance

## VI. Validation

- This task is concerned with ensuring that the gathered information in software meet certain standard of quality

## VII. Requirement Management

- Requirement management is defined as "the process of controlling and tracking changes in the requirements during the requirement engineering process of system development".
  - During requirement management, the project team performs a set of activities, identify, control and track requirements.
- Q. What is Feasibility Study? Explain its types also?

- ① Feasibility study is a study to reveal whether the project is feasible or not!
- ② To evaluate feasibility, a feasibility study is performed.
- ③ Feasibility study determines whether the solution considered to accomplish the requirements is practical and workable in software.
- ④ Feasibility study is important to find the answer of some questions like:
  1. Which resources or technology required to build a project.
  2. Do we receive profit from the project.
  3. Really project is worth it or not.
- ⑤ Following are the types of feasibility study:

## I. Technical Feasibility

- Technical Feasibility refers to the technical resources needed to develop, purchase, install or operate the system;
- Technical Feasibility check whether we have required technical resources to develop the project;
- A technical feasibility also analyse the technical skills & capabilities of technical team;

## II. Operational Feasibility

- Operational feasibility means that a proposed system will be used effectively after it has been developed;
- OF check whether project satisfies the client requirement or not;
- it checks the solution suggested by developer is acceptable.
- it analyzes whether users will adapt to new software.

## III. Economic Feasibility

- EF determines whether the required software is capable of gathering financial gains for an organisation.
- In EF study we study cost & benefits on the project.

- in this we separately identify cost of hardware, Software & development Part)
- It also analyze the project is beneficial for organisation or not.

Q. Explain Different Fact finding techniques in detail

- ① The success of any Project is depended upon the accuracy of available data accurate information can be collected with the help of certain techniques. These specific techniques for finding information of the system are called as fact finding Techniques.
- ② Fact finding techniques also called as information Gathering or data Collection.
- ③ Following are the different fact finding techniques:

### I. Interviews :- ( Explain Interviews )

- ① Interviews is the best method of fact finding.
- ② Interviews techniques is refer to the face to face communication between system analyst and user to gather information about problem.
- ③ interviews is the best method of producing qualitative information.
- ④ Interviews are strong medium to collect requirements.
- ⑤ During interviews , the client and analyst discuss with each other.
- ⑥ interviews is a question answer format to collect information.

- ① Interviews Can be formal or informal.
- ② Types of Interviews :-

## I Structured (closed) Interviews :-

- In structured Interviews, questions are already fixed. information gather is decided in advance.
- Advantage :-

- 1) it is easy for analyst to evaluate
- 2) These interviews follows Pattern.
- 3) These interviews required limited training.

### Disadvantages :-

- 1) High level of structure may not be suitable for all situations.
- 2) Cost of Preparation is High.

## II Unstructured (open) Interviews :-

- In these type question are not fixed and information gathered is not decided in advance , it is more flexible and less biased.
- Advantage :-

- 1) in this interviews, client gives the answer is free in their own words.

### Disadvantage :-

- 1) in this method, Some unwanted topic started during the interviews.

2) it may take Extra time to collect info

## II. Questionnaires :- (Explain Questionnaires)

Q Questionnaires technique Used to extract information from large number of people

② A questionnaire techniques is Used in survey format Because we have large number of People:-

③ In questionnaire techniques, a document with pre-defined set of objective questions and respective options is handed over to all customers to answer, which are collected and compiled;

### ④ Advantages :-

- i. This is cheapest Source of fact finding
- ii. Questionnaire can send to People by email or post.

### Disadvantage -

- i. In this technique, limited options are available to answer.

### ⑤ Example of questionnaires.

A Survey of college Using google form

### III. Record View -

- ① Records and reports are the collection of information and data about the system and its operations.
- ② This record view helps the analyst to get valuable information about the system and organization.
- ③ Record view easy to understand the system with actual in operations.
- ④ Example - Salary record, employee attendance record,

### IV. Observations -

- ① In this method, team of experts visit the customer's workplace.
- ② Analysts observe actual working of system.
- ③ They observe workflow of document at customer's end.
- ④ The experts team itself find some conclusions which help to form requirements expected from the software.
- \* The individual who performs the system investigation & who may or may not be related to computer programming is called as System Analyst.

## \* SRS (Software Requirements Specification)

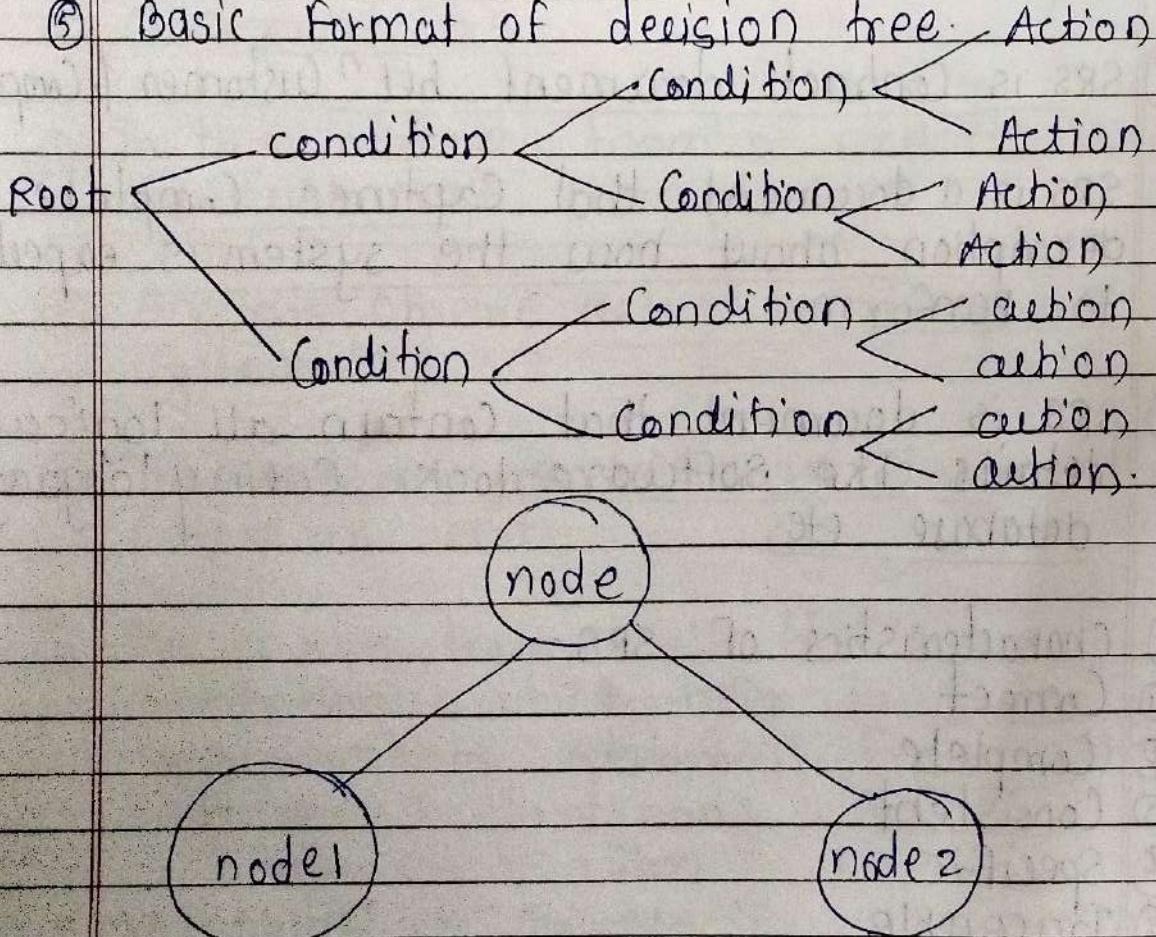
:-

- ① SRS stands for 'Software Requirement Specification'.
- ② SRS is a detailed description of Software System.
- ③ SRS is formal report of the Software.
- ④ SRS know the functional and non-functional requirement of Software.
- ⑤ SRS is Contract document bet<sup>n</sup> Customer & Company.
- ⑥ SRS is a document that Captures Complete description about how the system is expected to perform.
- ⑦ SRS is document that contain all logical details like Software look, Coding language, database etc.
- ⑧ Characteristics of SRS -
- ① Correct      Correct      Correct
- ② Complete      Complete      Complete
- ③ Consistent      Consistent      Consistent
- ④ Specific      specific      specific
- ⑤ Traceable      Traceable      Traceable

# 5. Analysis and Design Tools

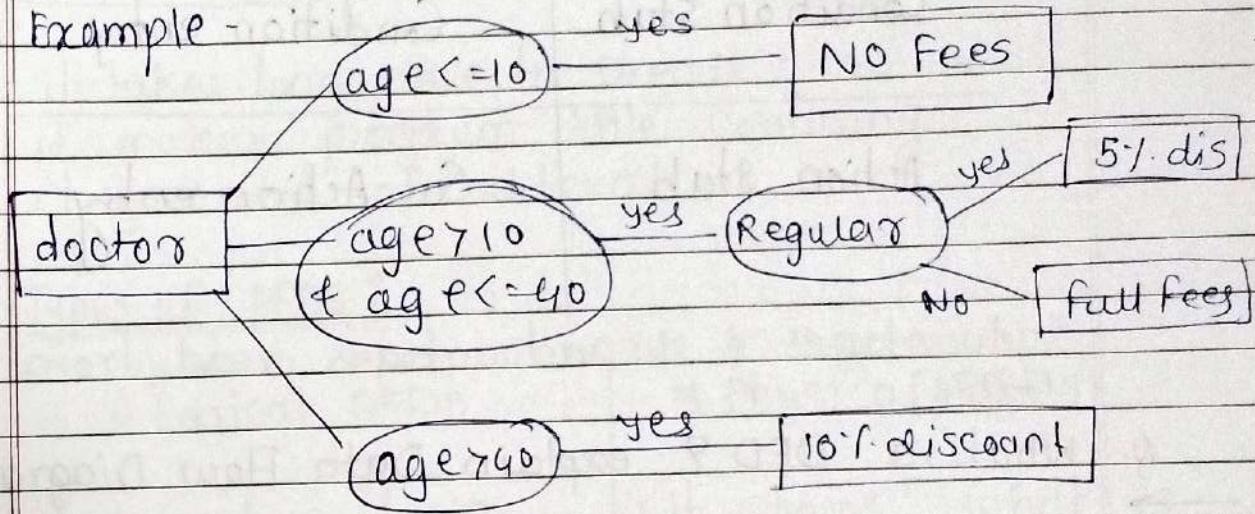
Q.1 What is decision Tree Explain?

- ① Decision Tree defined as 'a graphical representation of specific decision situations in a structured decision process'.
- ② Decision Tree represents Conditions and action sequentially.
- ③ a decision tree is a graph which uses as a branching method;
- ④ a decision tree diagram shows Condition and alternative actions.
- ⑤ Basic Format of decision tree



⑥ decision tree also known as decision diagram

⑦ Example -



⑧ Advantages -

- i. Decision trees are simple to understand.
- ii. It helps to identify actual decision made.
- iii. It is used to verify logics.

⑨ Disadvantages -

- i. Large decision tree can be hard to interpret.
- ii. Can have large errors.
- iii. Large number of branches can confuse.

## Q.2 Explain Decision Table or (DT).

- ① Decision table is a tabular representation of all conditions.
- ② We can derive decision table from decision tree.
- ③ It is used to describe the complex problem.
- ④ A decision table is defined as "a tabular representation used to describe and analyze decision situations".
- ⑤ Parts and components of decision table:

Condition stub

Condition entry

Action stub

Action entry

(advantages & disadvantages read from table)

Q. What is DFD? Explain Data Flow Diagram.

① DFD stands for Data Flow Diagram and also known as "Bubble chart".

② DFD shows the flow of data between various elements of a system.

③ DFD is an overview of what data system processes what transformation, what data are stored etc.

④ DFD is a good nature communication between user and system analyst.

⑤ Advantages -

① it represents system data in hierarchical manner.

② it focuses on process.

③ it explains logic behind the data flow.

④ Useful for communicating current system.

## ⑥ disadvantages:-

- ① it takes long time to create.
- ② it makes program little confusing.
- ③ it goes lot of alteration.

## ⑦ Types of DFDS :-

### 1) Logical DFDS

1. it shows "how it is going on".

2. it is more abstract in nature.

3. it is more logical format.

4. it shows clear idea what system is achieving.

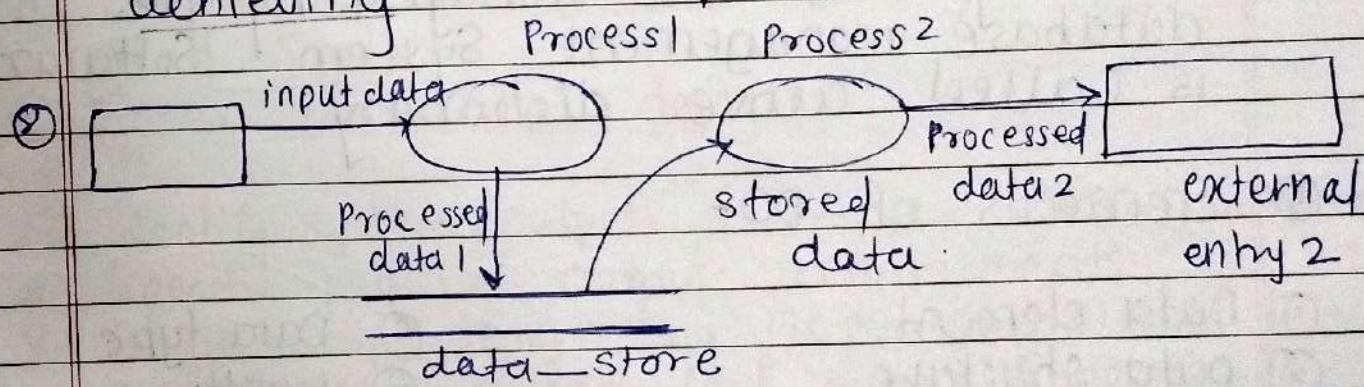
### 2) Physical DFDS.

1. it shows "what is going on".

2. it is more detailed in nature.

3. it is more implementation oriented.

4. it shows list for visualization.



⑧ Entity -        Process       data flow  

data store =

Basic Notation DFD.

## Q.4. What is data dictionary

- 
- ① Data dictionary defined as a Collection of descriptions of the data objects or items in System'.
  - ② A data dictionary contain metadata (data about data).
  - ③ A data dictionary stores an Organised Collection of information about data and their relationships, data-flows, data-types, data stores, processes and so on.
  - ④ Types of data dictionary :

I. Passive dictionary - if the user data dictionary is being used by the Users designers and the database administrators then the data dictionary is called as Passive dictionary

II. Active dictionary - Data dictionary being directly used and managed by the database management system Software is called active dictionary

### ⑤ Elements of DD-

- ① Data element
- ② Data structure
- ③ Data Flows and data stores
- ④ Data Names
- ⑤ Alias
- ⑥ Data type
- ⑦ length
- ⑧ Data value
- ⑨ data Description
- ⑩ Adv & disadv read from book

Q. What is Input Design and process of input design

- ① Input design means designing the Screens Used to enter the information.
- ② Input design facilities the entry of data into Computer system
- ③ Process of Input design -

1. Identify System inputs and review logical requirements.
2. select appropriate GUI Controls.
3. Design, Validate and test inputs Using Some Combination of i) layout tools ( hand sketching ) ii. Prototyping tools ( spreadsheets )
4. if necessary, design the Source document.

Q. What is Output design and process of output design.

- ① Output design involves Specifying how production of on-screen reports.
- ② Output design marks the beginning of a good System.
- ③ Process of Output design -

1. Identify system outputs & review logical requirements.
2. Specify physical output requirements.
3. As necessary, design any pre-printed external forms.
4. Design, Validate and test output Using Some Combination of a layout tools ( Sketches etc ) b. Prototyping tools ( Spreadsheets ) c. Code generating ( report writer )

## Q. What is PSEUDOCODE ?

- ① Pseudocode is made up of two words 'pseudo' and 'code'.  
Pseudo means imitation or false  
Code means instructions, written in programming language.
- ② Pseudo means the imitation of code means the instruction written in a programming language.
- ③ a ~~pe~~ pseudocode is a readable description of what an algorithm or program must do.
- ④ Pseudo code is a simply way to communicate specifications to programmers or users.
- ⑤ advantages

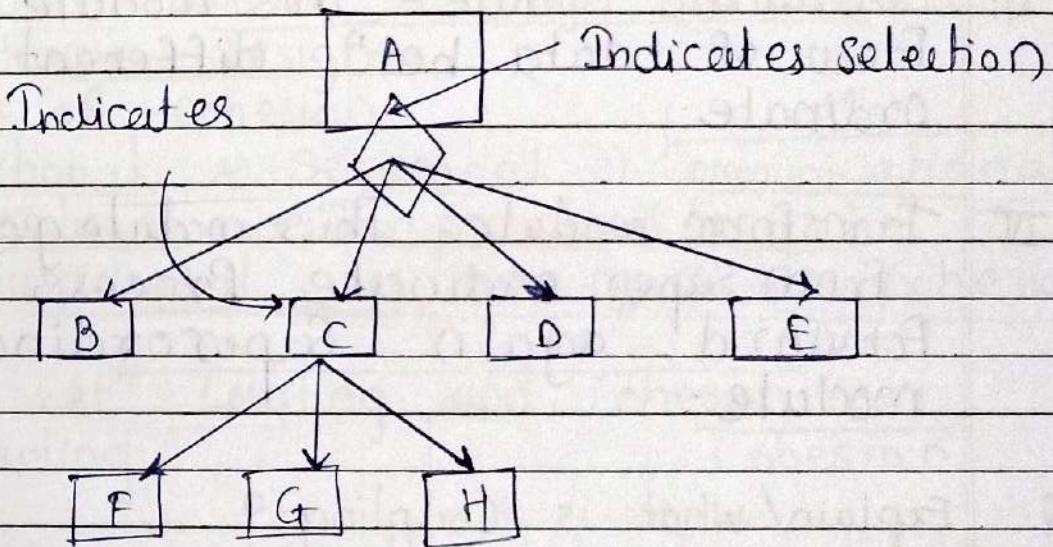
1. Writing of Pseudocode much less time & effort.
2. Pseudocode reduces Complexity.
3. Increased flexibility.

## ② disadvantages

- a. In this Graphic representation is not available.
  - b. There are no standard rules in pseudocode.
- ↑ Structured design - it is disciplined approach to Computer System, it deals with size & complexity of program it is developed by Ed Yourdon and Larry Constantine

Q. Write short Note on Structure chart.

- 
- ① structure chart is a hierarchy diagram.
- ② structure charts of a Program is a graphic representation of its structure.
- ③ structure chart do not show internal procedure.
- ④ structure charts are good for designing large and complex programs.
- ⑤ Format of structure chart :-



Q. Explain/ what is module and its types?

- ① Module is a logically separable part of program which is discrete & identifiable with respect to compile and load.
- ② Types of module : There are 4 types.
  - 1) Afferent module
  - 2) Efferent module
  - 3) Coordinate module
  - 4) Transform module

I. Afferent Module - This module get data from Sub-ordinate & forward it to super-ordinate module this is called input module or afferent module.

II. Efferent Module - This module get data from super-ordinate and forward it to sub-ordinate module this is called output module or efferent module.

III. Co-ordinate Module - This module manage flow of data bet" different super-ordinate.

IV. Transform module - This module get data from super-ordinate process it forward again superordinate module;

Q. Explain/ what is Coupling ?

- ① Coupling refers to the interdependencies between modules.
- ② We represent Coupling in "Top level Design"
- ③ It also represents interaction between modules of s/w
- ④ Coupling Should be low for better software design.
- ⑤ Types of Coupling -

- 1) Normal Coupling
- 2) Data Coupling
- 3) Stamp Coupling
- 4) Control Coupling

### Q. Explain Cohesion.

- ① A cohesion represent the detailed design.
- ② It describes how elements are closely related to each other in module.
- ③ Cohesion should be high for better Software design.
- ④ Types of cohesion:
  - 1) functional 2) Sequential 3) Communicational
  - 4) Procedural 5) Temporal 6) logical
  - 7) Coincidental are the types of cohesion

### Q. DIFF' bet<sup>n</sup> Coupling and Cohesion.

#### Cohesion .

- | Coupling  | Cohesion   |
|---|--|
| 1. it is inter-module Concept                               | 1. it is intra-module Concept.                         |
| 2. Coupling shows the relationship bet <sup>n</sup> modules | 2. Cohesion shows the relationship within module       |
| 3. Coupling should be low for better software design        | 3. Cohesion should be high for better software design. |
| 4. In Coupling, modules are linked to the other modules     | 4. In Cohesion, the modules focuses on single thing.   |

# 6. Software Testing

Q. What is software testing.

→ Slw testing is an imp part of Os/w development.

- ① Software testing is a technique of Software to find the error or mistake in a particular Software design.
- ② Software testing is the process of executing a software and program or an application for finding the bugs.

+ Need of Software testing.

- ① Software testing is required to point out the defects and errors.
- ② Software testing ensure the quality of product.
- ③ Software testing is required for an effective performance of slw application.

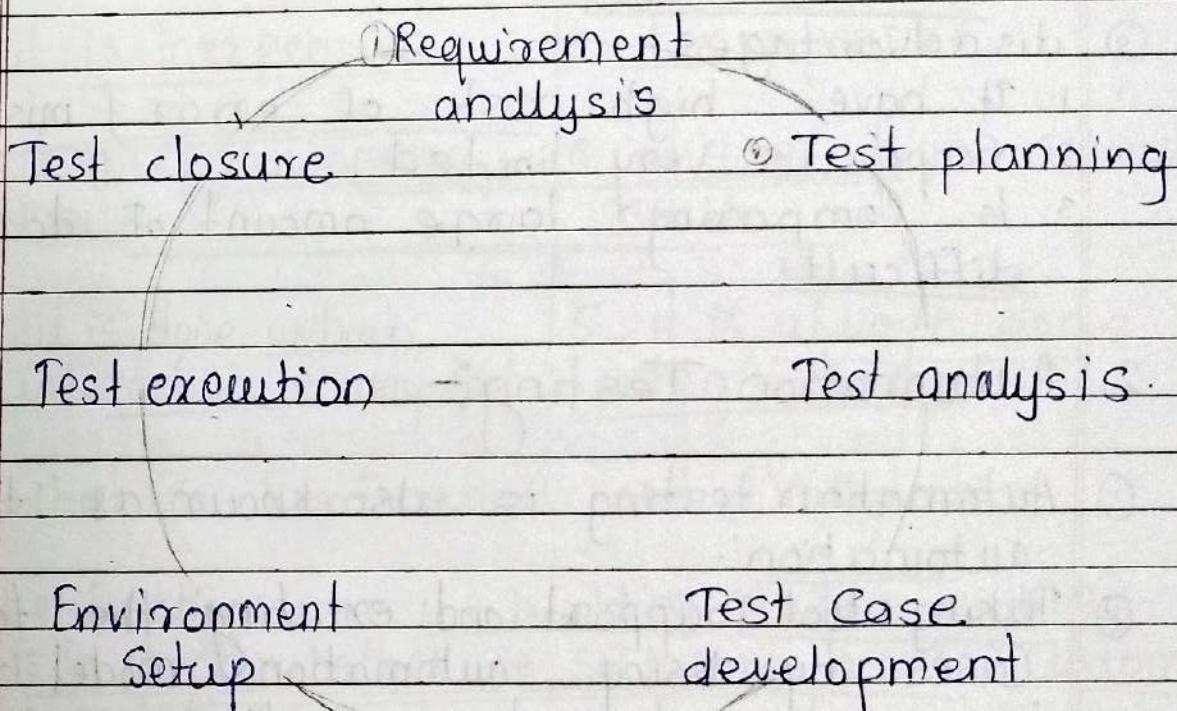
+ Objectives and Principles of slw testing.

- ① Prevent defect
- ② To make sure that at end results meets the business requirements.
- ③ To ensure that it satisfies the BRS and SRS.
- ④ To provide quality product.

\* PDCA Cycle - it is Project Planning Tool  
it include Plan, Do, check, Act. it also known as Deming Circle / wheel.

## Q. Life Cycle of Software Testing

- ① Software testing life cycle (STLC) is the process of Software testing in which Specific steps to be executed in a definite Sequence to ensure that the quality goals have been met.
- ② Phases of STLC :-



(Explanation read from textbook:-)

## # Types of Testing :-

1. Manual Testing
2. Automation Testing

Q. Explain different types of testing.

## 1. Manual Testing:-

- ① Manual testing means testing a software manually i.e. without using any automated tools or any script.
- ② advantages-
  - 1. it is cost effective for test.
  - 2. no need of programming knowledge.
  - 3. Cost is less.
- ③ disadvantages-
  - 1. It have high risk of error & mistake.
  - 2. Scope is very limited.
  - 3. To Comparing large amount of data is difficult.

## 2. Automation Testing:-

- ① Automation testing is also known as 'test automation'.
- ② Taking tool support and executing the test cases by using automation model tool is called automation testing.
- ③ advantages-
  - 1. Fast
  - 2. Repetable
  - 3. Reusable
  - 4. Programmable
  - 5. Comprehensive
- ④ disadvantages-
  - 1. Expensive Tool
  - 2. Tools skill take time
  - 3. Setup problems

## \* DIFF' bet' Manual Testing & Automation Testing.

### Manual Testing

1. It is executed by human

2. More testers required

3. it is inexpensive

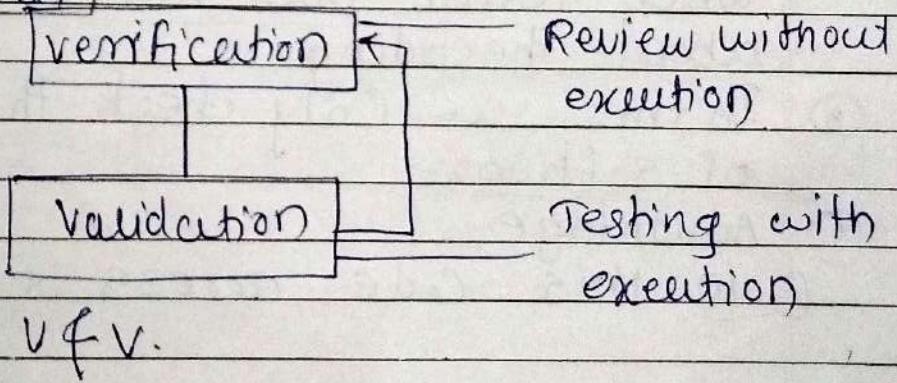
4. it gives low accuracy result.

5. it is done without interaction of any tool

Q. What is Verification and Validation.

① Verification (are we building product right?) -  
is to check whether software meets customer expectations.

② Validation (are we building the right product) -  
is to check whether software confirms to specification,



## Q. Diff' bet' Verification & Validation

### Verification

### Validation

- |  |  |
|--|--|
| 1. Verification is doing things right          | 1. Validation is doing right things.           |
| 2. it is product based                         | 2. it is product based.                        |
| 3. it ensures Software meets all functionality | 3. it ensure functionality meets its behavior. |
| 4. Verification take place first               | 4. Validation occurs after Verification        |
| 5. Done by developers                          | 5. Done by testers.                            |

Q. Explain w Black Box testing and white box testing.

### I. Black Box Testing -

- ① It is also known as Behavioral Testing
- ② Black Box testing is that testing in which tester does not have knowledge about the coding
- ③ In this, we Only check the functionality of Software.
- ④ Advantage.
- ⑤ in this code access is not required

- This testing type suited for large code.
- It is clearly separates users view from developer view.
- ⑤ Disadvantage :-
  - ① it is very inefficient testing
  - ② Test cases are difficult to design
  - ③ it has limited coverage.

## II. White Box testing -

- ① White Box testing also known as structural testing or Glass box testing.
- ② In white box testing tester has knowledge of coding & internal structure of existing Software.
- ③ In this testing we check structure based testing technique of Software.
- ④ Advantage.
  - 1. It helps to optimizing the code.
  - 2. This is very easy & simple testing.
  - 3. In this testing tester remove the extra line code.
  - 4. White box is easy to automate.

### ⑤ disadvantage.

- 1. It is difficult to maintain white box testing.
- 2. In this testing skilled tester is needed.

Q. DIFF<sup>n</sup> Bet<sup>n</sup> white and Black Box testing.

### Black Box testing

### White Box testing

- |   |  |
|---|--|
| 1. It is also known as behavioral or functional, closed box testing | 1. It is also known as, clear-box and structural testing |
| 2. Tester does not have knowledge about code                        | 2. Tester has knowledge about code;                      |
| 3. it is done by end-users, developers, testers                     | 3. it is done by testers & developers.                   |
| 4. it is exhaustive and least time consuming                        | 4. it is more exhaustive & time consuming                |
| 5. Not suited for algorithm testing                                 | 5. Suited for algorithm testing                          |
| 6. it does not require knowledgeable person                         | 6. it requires knowledgeable person.                     |

Q. Explain Unit testing

- ① Unit testing is a process to check the independent and individual unit or module of software
- ② it is first level of functional testing

③ Unit testing could be white-box or black-box both

④ The goal of Unit testing is to isolate each part of the program and show that individual parts are correct in terms of requirements and functionality.

⑤ advantages-

1. It helps to reduce cost of bug fixes
2. It helps in simplifying debugging process
3. also helps in maintain and changing code

⑥ disadvantages -

1. It cannot catch each & every bug in s/w appl'
2. It cannot evaluate every execution part
3. limit to no. of scenarios and test data

⑦ JUnit, NUnit, PHPUnit, JMockit, EMMA are the some Unit testing Tools.

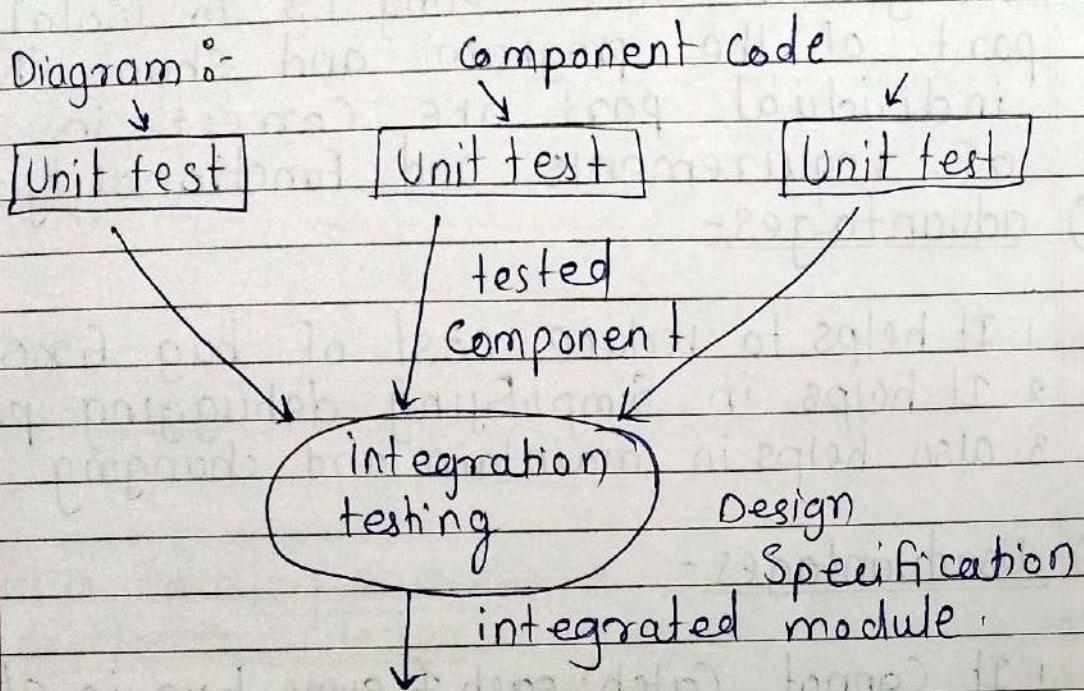
Q. Explain / what is Integration Testing?

① Integration testing is defined as "the testing of combined parts of an application to determine if they function correctly".

② Integration testing will integrate all module one by one & check the compatibility issue.

③ Integration testing defines relationship between Components

④ Diagram :-



⑤ Types of Integration Testing

- 1. Top down Integration } explanation
- 2. Bottom - up Integration. } read from text
- 3. Sandwich Integration } book }

Q. What is System testing ?

① System testing defined as "a testing conducted on a complete integrated system to evaluate system compliance with its specified requirements".

② System testing, we put the SWI

in different environment and check whether software is compatible with new environment or not.

- ③ This testing is performed by tester
- ④ Advantages -

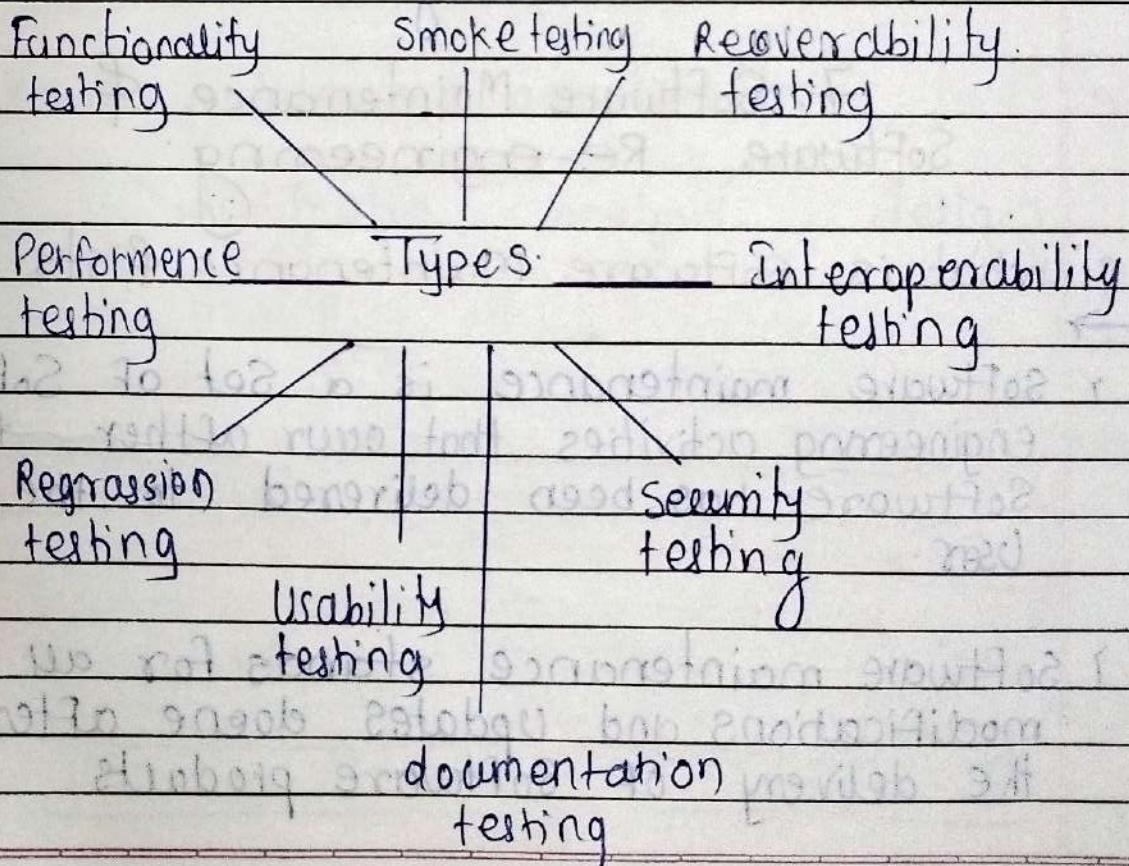
1. it specify how application should behave
2. can be run automatically

- ⑤ Disadvantages

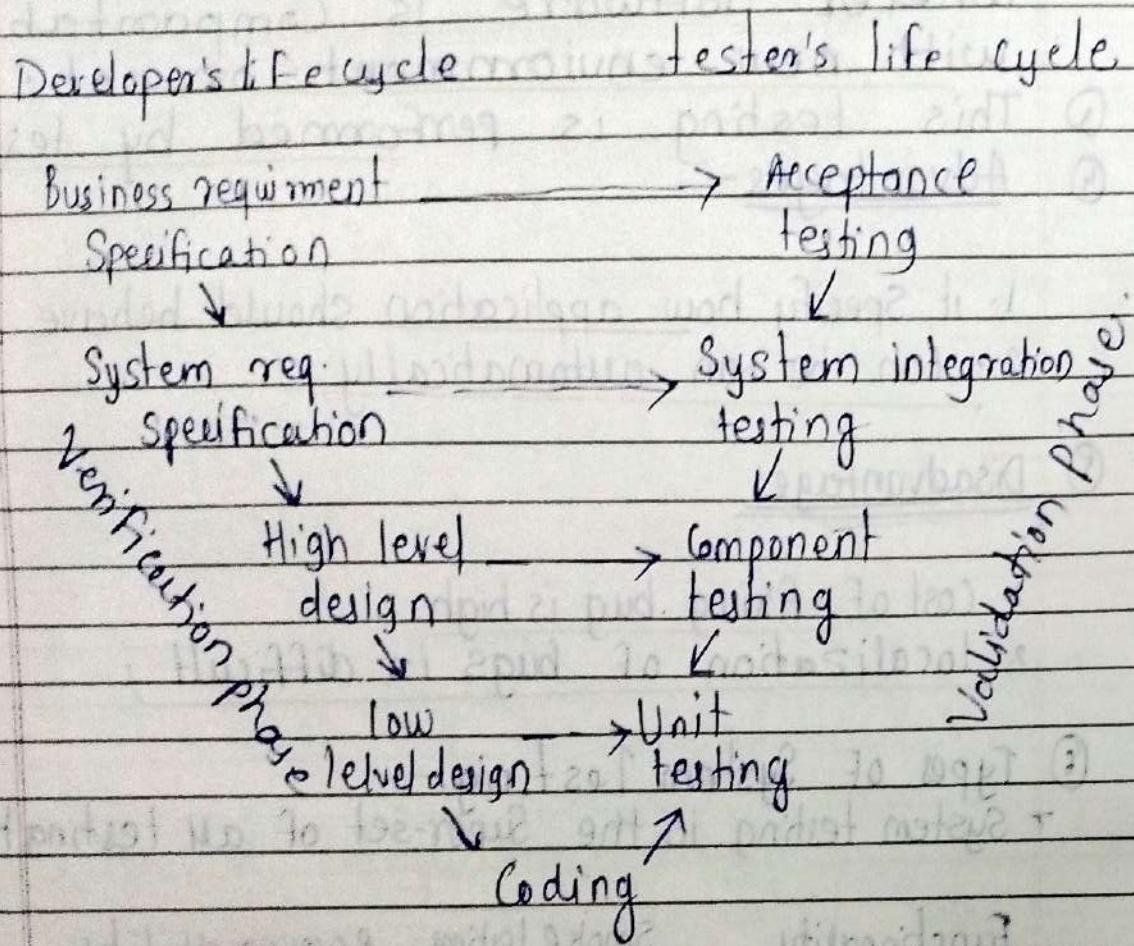
1. cost of fixing bug is high
2. localization of bugs is difficult

- ⑥ Types of System Testing -

\* System testing is the Super-set of all testing type



## \* Diagram of V and V model



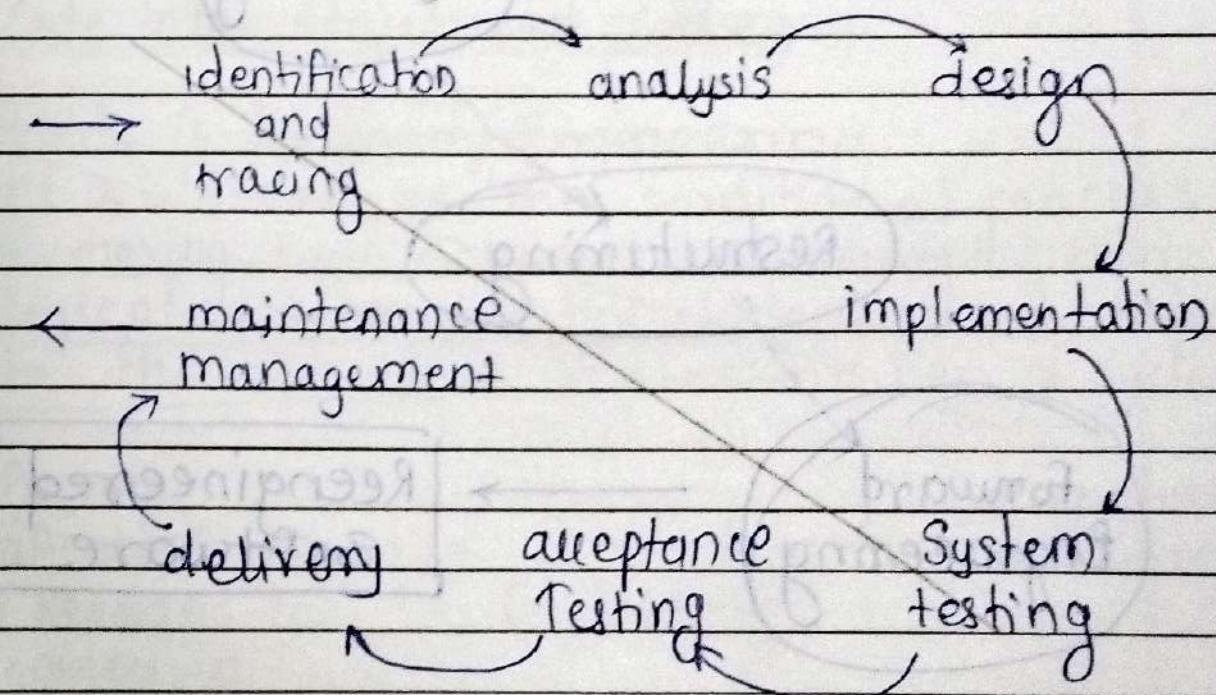
## 7. Software Maintenance & Software Re-engineering

- Q What is Software Maintenance & its types?
- Software maintenance is a set of software engineering activities that occur after the software has been delivered to the end-user.
- Software maintenance stands for all the modifications and updates done after the delivery of software products.

- > Software maintenance is defined as "the process of modifying a software system after delivery to correct faults, improve performance."
- > Maintenance keep Software Up-to-date with environment change.
- > Y2K, operating System patching, Anti-virus Software are the some examples of Software maintenance.
- > Types of maintenance :-

  - ① Corrective maintenance
  - ② Adaptive maintenance
  - ③ Perfective maintenance
  - ④ Preventive maintenance

- > Various maintenance activities :-



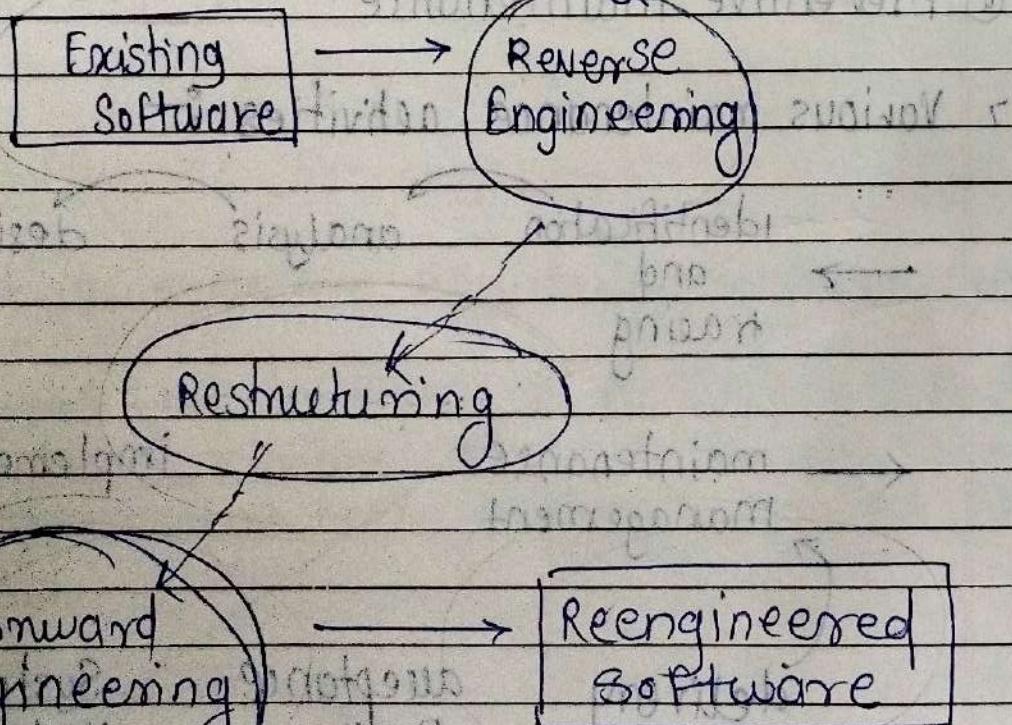
## Q. What is Software Reengineering

→ Software Reengineering is technique refers to the systematic transformation of the present software system into a new form to make quality performance of improvement.

→ activities of Software reengineering :-

- 1) decide
- 2) Perform
- 3) Re-structure
- 4) Re-structuring tuning
- 5) Apply forward Engineering

→ Process of Reengineering



Q. What is Reverse engineering.

→ Reverse engineering is defined as "the process of analysing a Subject System to identify the System Components and their interrelationships and create representation of the System in another form at a higher level of abstraction".

Existing Software → Reverse Engineering → Program Specification

Q. Restructuring

→ it is defined as "the Process of transforming an existing System from one form to another without changing the logic or functionality".

→ it involves transformation of Unstructured Code into Structured code.

Q. What is Forward Engineering

→ It is defined as "the traditional process of moving from high level abstractions and logical implementation-independent designs to the physical implementation of system".

② Program Specifications → Forward Engineering → Reengineered software  
From Reverse engineering

Process of forward Engineering