Software Engineering

Unit 1

Contents :-

- → Introduction of Software Engineering.
- → Software Components.
- → Software Crisis.
- → Software engineering Process.
- → Software Quality attributes.
- → SDLC - Types of Models.

Introduction of Software :-

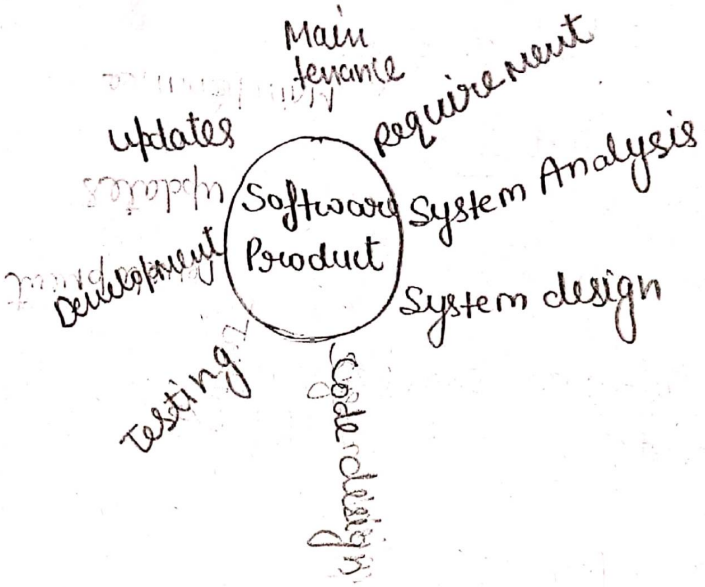
Software :- "Software engineering is the practical application of scientific knowledge in the design and the construction of the programs and associated documentation required to develop, operate and maintain them."

Software is more than just a program code. A program is an executable code, which services some computational purpose.

→ Software is considered to be collection of executable programming code, associated libraries and documentation

→ Software, when made for a specific requirement is called Software product.

Engineering :- Engineering is all about developing products using well defined, scientific, principles and methods.



Software Engineering :- Software engineering branch associated with development of software product using well defined scientific principles, Methods and procedure

The outcomes of Software Engineering is an efficient and reliable Software product.

Need of Software Engineering :-

→ Large software :- it is easier to build a wall than a house or building likewise, the size of the software become large.

* Scalability :- If the Software process were not based on scientific and engineering concepts, it would be easier to created new Software than scale an existing one.

* Cost :- The cost of the software remains high if proper process is not adaptable.

* Quality Management :- Better process of s/w development provides better and quality s/w product.

* Dynamic Nature :- Always growing and adapting nature of the s/w highly depends upon the environment in which the user works.

⑥ Portability :-

→ Software can be transferred from one computer system or enviroment to another.

⑦ Adaptability :- Software allow differing System constraints and user needs to be satisfied by making changes to the Software.

* Program VS Software Product :-

1. A program is a set of instruction which is given to a computer in order to achieve a specific task whereas a Software is when a program is made available for comercial busines and is properly documented along with its licencing.

   Software = Program + documentation + licencing

2. A program is some one of the stage involved in the development of the Software, a software development usually follow a life cycle.

Software Engineering :-

Introduction :-

*Software :- Software is a program or set of program containing construction instruction which provides desired function -ality.

Engineering :- Engineering is the process of designing and building something that services a particular purpose and a cost effective solution to problems.

Software Engineering :-

→ Software Engineering is a systematic approach to the design development operation and maintenance of a software system.

* Dual Role of S/w :-

1. As a product :-

→ it delivers the computing potential across network of hardware.

→ it acts as information transfer because it produces, manage acquires, modifies, displays or transmits information.

2 :- As a vehicle for delivering a product :-

→ It provides System functionality (e.g. payroll System)
→ it control other Software (e.g. an operating System)
→ it helps build other Software (e.g. Software tools)

Objectives of Software Engineering :-

① **Maintainability :-**
→ it should be a feature feasible for the software to evolve to meet changing requirements.

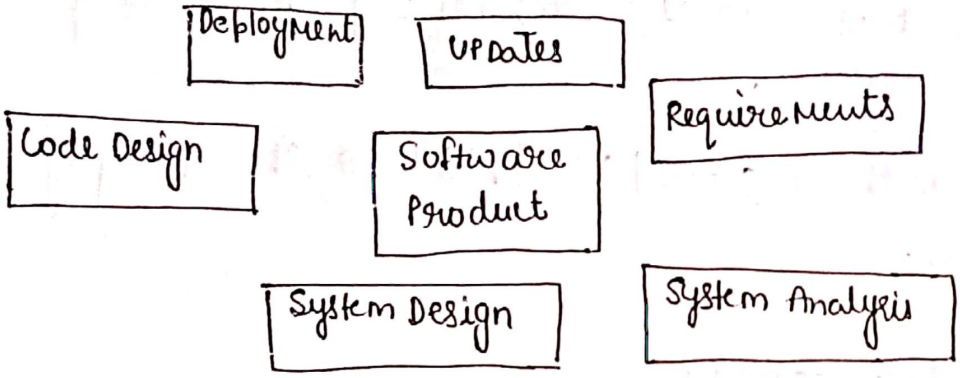
② **correctness :-**
→ A Software product is correct, if the different Requirement as specified in the SRS document have been correctly implemented.

③ **Reusability :-** A S/w product has good reusability. if the different Modules of the product can be easily be reused to develop new products.

④ **Testability :-** Software facilities both the establishment of test criteria and the evaluation of the Software with respect to those criteria.

⑤ **Reliability :-** In this case, Software can be transferred from one computer System or environment to another.

Why Slw engineering is popular ?



① Large Slw :-
② Scalability
③ Adaptability
④ Cost
⑤ dynamic Nature
⑥ Quality Management.

Relationship of Slw Engineering with other disciplines

How Software engineering related to other decipline:

Computer Science :- Give the scientific foundation for the Slw as electrical engineering mainly depends on Physics.

Management Scence :- Software engineering is labor-intensive work which demands both technical and managerial control.

Economics :- Software is a component of a much larger System. Slw engineering helps you in resources estimation and

and cost control.

System Engineering :- Most Slw is a component of a much larger System.

→ Slw engineering methods should be applied to the study of this type of system.

*Challenges of Slw Engineering :-

→ Here are some critical challenges faced by Slw engineering.
→ The diversity of Slw System should be communicating with other.
→ Dealing with the increased complexity of Software need for new application.
→ Increased market demands for turnaround time.

*Attributes for Slw products :-

→ The characteristics of any Slw product include feature ~~with~~ which are displayed by the product when it is installed and put in use.

→

Characteristics of Good SW :-

→ A software product can be judged by what it offers and how well it can be used.
This SW must satisfy on the following grounds :-

→ Operational
→ Transistional
→ Maintanance

1. Operational :- This tell us how well the software works in operations. it can be measured on.

→ Budget
→ Usability
→ Efficiency
→ Correctness
→ functionaitiy
→ Dependability
→ Security
→ Safety

2. Transistional :-
This aspect is important when the SW is moved from one platform to another.

→ Portability
→ Interoperatibility
→ Reusability
→ Adaptability

3. Maintanance :- How well the SW has the capability to maintain itself in the ever-changing environment.

1 → Modularity
2 → Maintenability
3 → flexibility
4 → Scalebity

* Component of Software :-

→ Software Engineering is concerned with all aspect of Com - puter based development including hardware, Slw, and process engineering.

1. SET OF PROGRAM
2. SOFTWARE DOCUMENTS

Both are the main component of Slw.

→ A program is a set of instruction which discovered and developed by individuals to solve their days to day simple mathmatical and logical problem.

* SOFTWARE CRISIS :-

* Software crisis :-

→ Software industry is a very fast growing dynamic industry, each new day comes with some new invention and introducing new technology.

→ This development force slw developer to keep watch a new Technology and invention which makes the work of slw developer critical and complex and causes the slw crisis.

→ In other words we can say - Software crises is the set of diffi culties and problem encountered while developing slw.

(6)

* Software Quality Attributes :-

→ "Conformance to explicit stated functional and performance requirements, explicitly documented development standard and implicit characteristics that are expected of all professionally developed software."

→ Attributes of Software are :-

→ Correctness :- A System is functionally correct if it behave according to its functional Requirements.

→ Reliability :- A System is said to reliable if it gives desired output even in case of change of components.

→ Portable :- A Software System portability is the ease with which a Software System can be adapted to run on computer other than one for which it was design

→ Efficiency :- It is the ability of s/w system to fulfil its purpose with best possible utilization of necessary Resources such as a time, transmission channel.

* Characteristics of Software :-

There are various characteristic of software including :-

→ **Understanability :-** To what extent is the process explicity defined and how easy is to under - stand the process.

→ **Visibility :-** Do the process activities culminate is clear result so that the progress of process is visible.

→ **Robustness :-** Can the process continues in spite of unexpected problems

→ **Acceptability :-** Is the defined process acceptable to and usable by engineering responsible for producing software product.

→ **Supportability :-** To what extent CASE tool support the process activities.

5) Integration & System testing
6) Delivery, Implementation and Maintenance

(8)

* Requirement Analysis :-

→ This step is defined as to identify and document the exact requirement of the software which can be fulfilled with the initial step of communication between the customer and the developer.

→ This step is the first and is defined as a identify and document the exact requirement of software which can be fulfilled with the initial step of communication b/w the customer and the developer.

*2. Design :- with full and complete understanding of the requirement analysis phase, the next step is the designing of the s/w.

→ This step is divided into two level designing named as
① Preliminary Design or High level Design
② Detailed Design or Low level Design.

① Preliminary design :- The brief overview of the s/w architecture and structure ratherthan goes into the detailed in detailed design of the module.

③ Coding :- This is the phase that purpose produced the actual code that will be delivered to the customer the operational product.

* Software Development life cycle (SWDLC) :-

→ A Software Development life cycle (SWDLC) is an abstract representation of gradual development and evolution of Software.

→ SWDLC is a Series of Sequential or concurrent Steps of the Software development process.

→ It is a diagrammatically representation which also provide description of various phases and their sequence in life cycle of SW product.

⇒

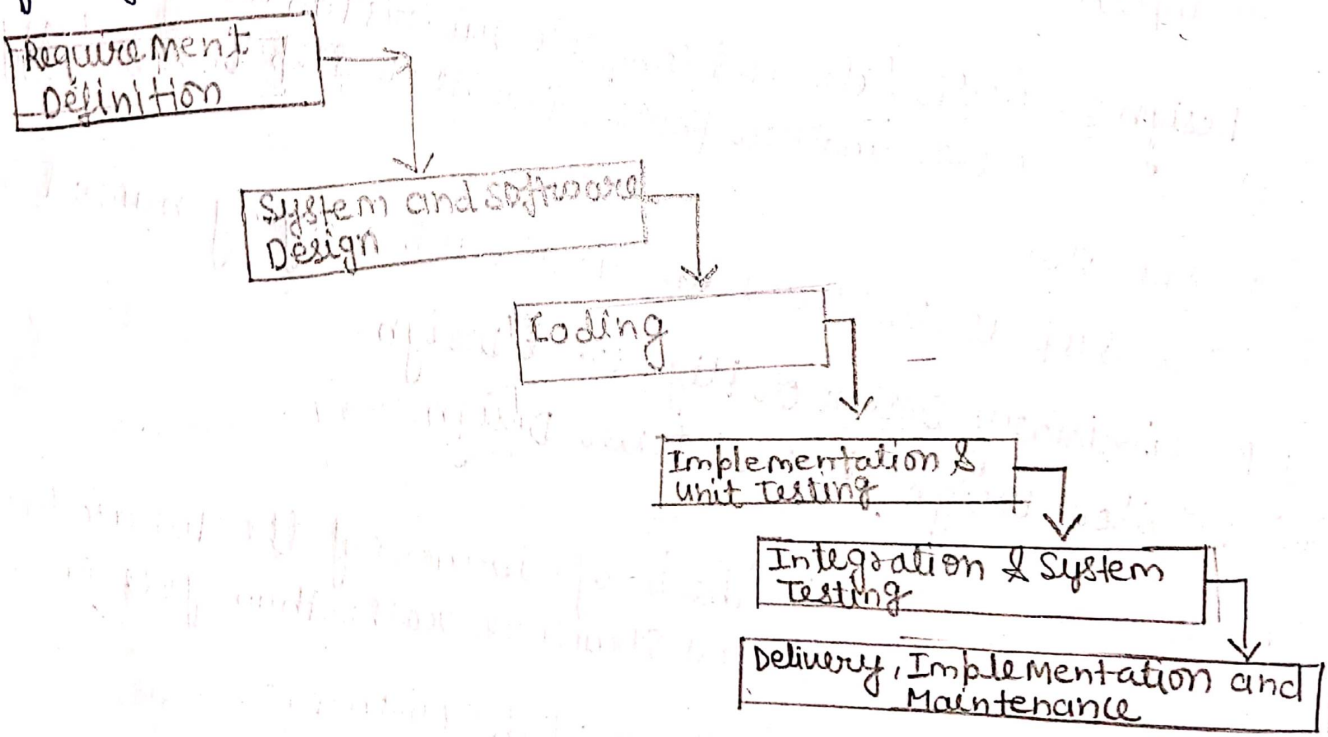


Figure :- Phases of SWDLC

1. Requirement Analysis
2. System and Software Design
3. Coding
4. Implementation & Unit Testing

* Functional Requirements :-

→ describe functionality of System services
→ depends on type of s/w, expected users.

* Non functional Requirements :-

→ These define System properties & constraints
Ex :- Reliability, Responsible time & storage Requirements.

* Functional Requirement Specification :-

→ In one sentence "what the System should do."
→ This is the list of the actual services which a System will provide.
→ This is the list of the services / functions which a user wants from the s/w.

examples :-

→ features of s/w which the client demands.
→ Business rules of the particular organization for which develop ment s/w.

* Non functional Requirements :-

→ How a System should behave while performing the operation.
→ There are the constraints on the services which System is offering. e.g. Timing of operation way to response in parti - cular condition.
→ for examples :-
1) Recoverability
2) Response Time
3) Example :- Database get ready after update within 2 second.

Types of SDLC :-

There are many Software Engineering (SDLC) SW development life cycle model as there is no general agreement about various phases and their sequence in product life cycle.

→ Some most commonly used life cycle model are given:

1. Waterfall Model
2. Prototype Model
3. Spiral Model
4. Evolutionary development Model
5. Iterative Enhance Model

*Waterfall Model :-

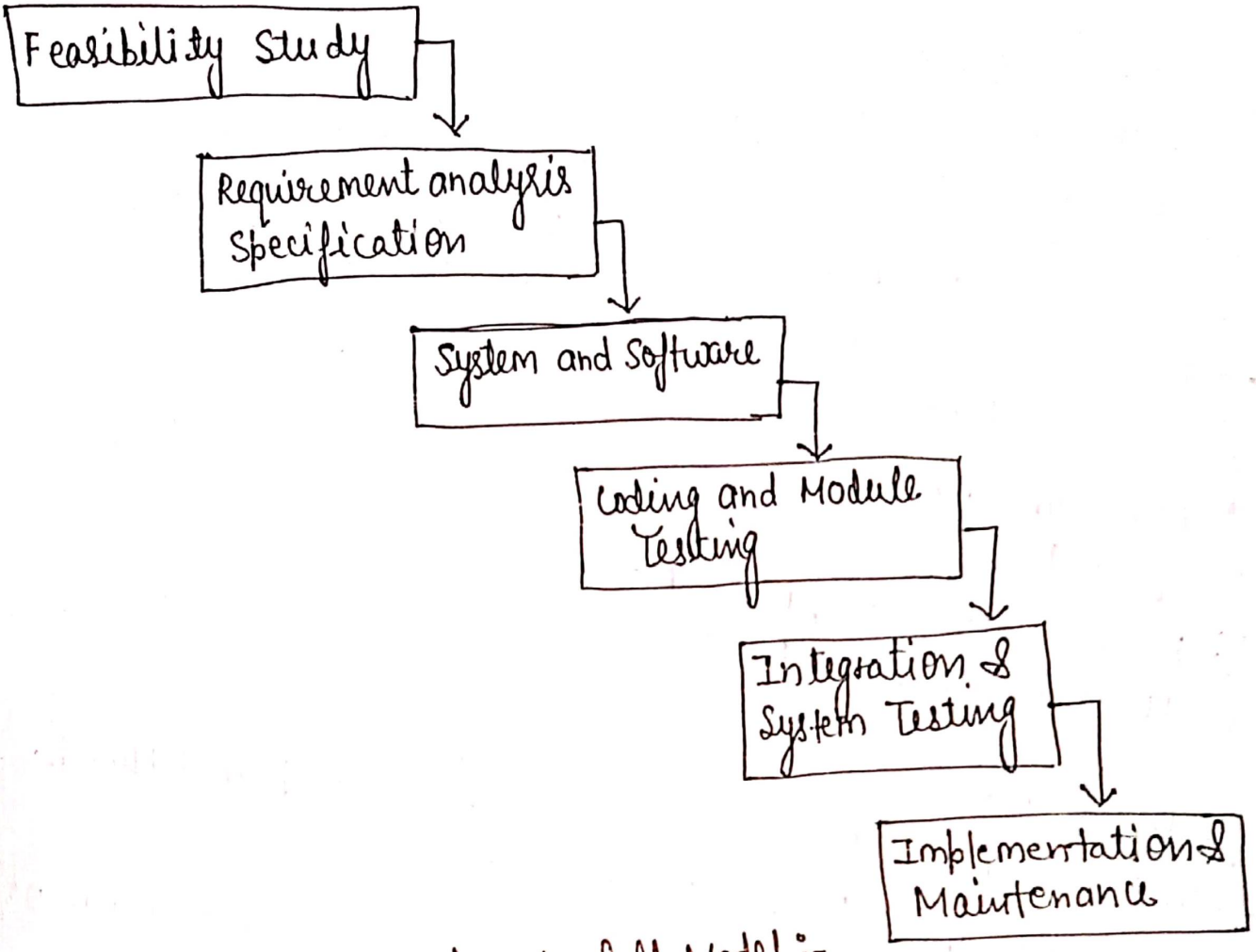
→ Waterfall Model is a theoretical SW development Model.

→ it is developed by BOEHM in 1970.

→ It is also known as classical, traditional, conventional or linear Segment model.

→ There are different stages to the development and the output of first stage flow to the next stage and so on. It force on

→ Each phase of this model is well define the starting and ending criteria which is to be documented by which the starting and ending criteria which is to be which the standard output produce by each pase can formulate.



* Classification of phases of waterfall Model :-

Feasibility Study :- This phase is used to check whether the new proposed system is economically, technically Operationally feasible or not.

② Requirement Analysis & Specification :-

→ This phase give specification about what is the system for.

→ It analyze and specifies the requirement of customer and document them properly.

(11)

* System & Software Deriging :- In design phase over all structu or architecture is developed which is transformation of requirement specified in SRS.

There are two main types of design approach:

→ Traditional design Approach
→ Object oriented design.

Coding & Module Testing :-

→ In this phase System design is translated into source code also called program code. The programming for different module is done in Selected programming language.

→

Integration & System Testing :- Individually tested modules integreated have according to planned System to develop the System.

→ There are two main testing:
→ Alpha Testing
→ Beta Testing

*Implementation & maintenance :- In this phase system is installed at the user end and it is checked if there is any upgradation required in hardware and s/w element at user end as per our s/w so that is made available.

* Prototype Model :-

→ The prototype model firstly a working prototype developed instead of developing actual software.

→ This developed according to available requirements which basically have limited functions, low reliability while it passed through all stages of development.

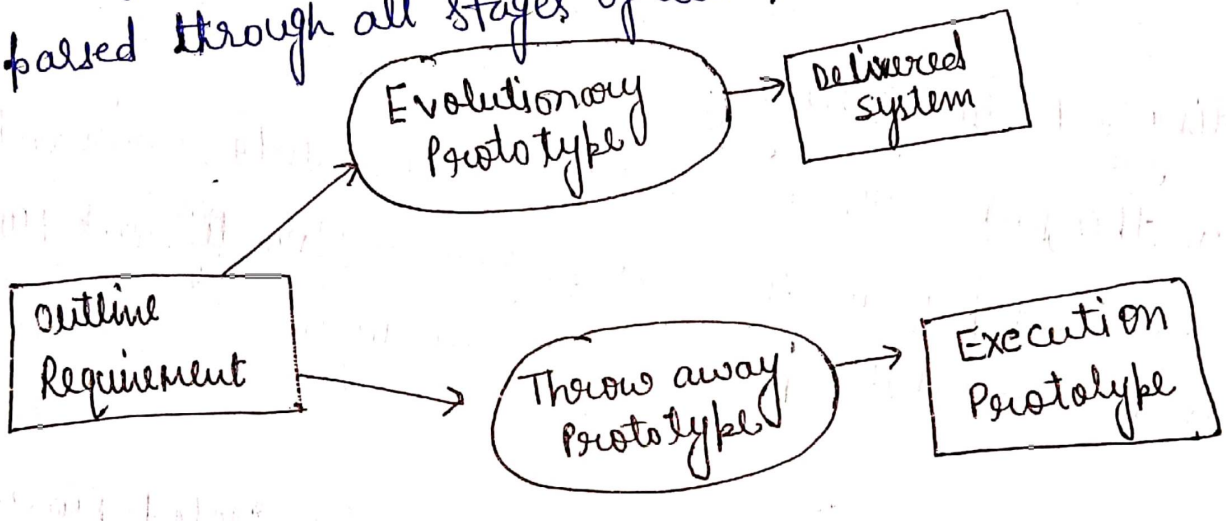


fig:- Prototype Model

① Evolutionary Prototype :- In this type of prototyping the main objectives is to work with user. to explore their requirement and delivered them a final system.

→ Gather the requirements.
→ Develop a working prototype on basis of initial requirement.
→ Handover to user to the working prototype feedback.
→ Make changes and add new requirement according according to user feedback in working prototype.

(13)

Throw-away prototype :- Throw-away prototype is concentrate on less well understood requirement of user.

→

* Spiral Model :-

→ The Spiral Model is based on the evolutionary approach. Proposed by Boehm.

→ This Model attempts to use the linear, Sequential and Stepwise procedure of linear Sequential Model and also

→ This Model is represent the form of spiral where each spiral is connected to another spiral to represented in the various Phase.

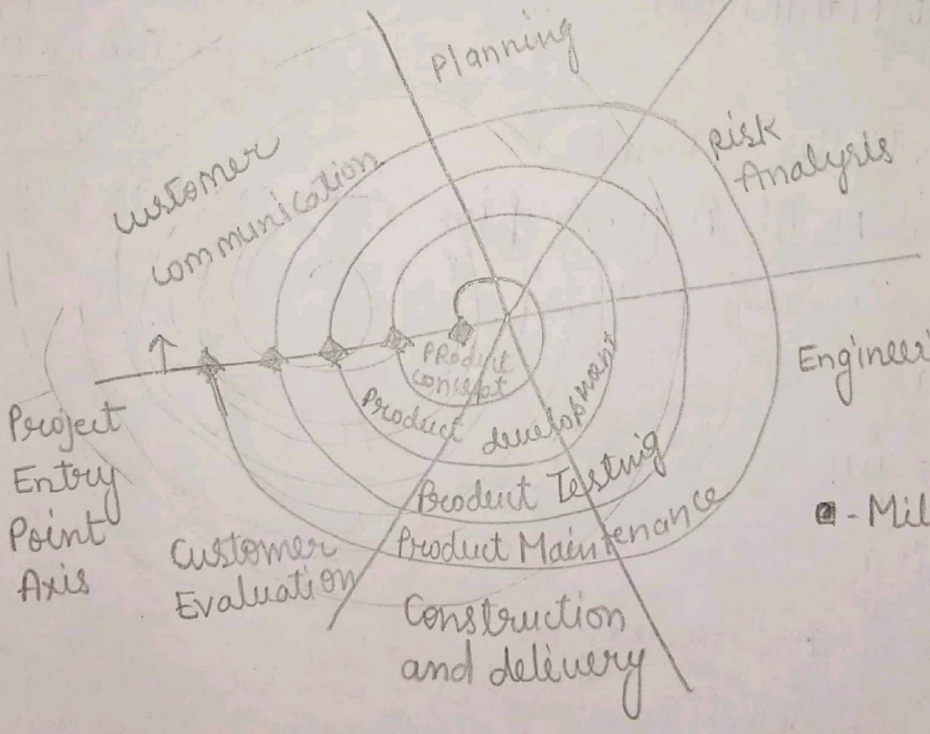


fig:- Spiral Model

→ Each iteration in the spiral shows a phase of the software process.

(14)

→ The process begins at the center of the spiral in clock wise direction.

→ There are totab of four loops and Spirls.

1. Product concept :- Product concept does the feasibility Study and produced the product specifi - cation.

2. Product development :- Product development develops the Software according to the maintione Specification.

3. Product Testing :- It includes the testing of product.

4. Product Maintenance :- n it maintenance the product where the changes can be done in the controller Manner.

→ a Spiral Model is Particifion into six task regions

1. Customer Communi cation
2. Planning
3. Risk Analyeis
4. Engineering
5. Construction and delivery
6. Customer Evaluation

(15)

Advantages of Spiral Model :-

→ The main and most important feature of this Model is Risk Analysis. The risk analyst will try to reduced the task risk arised.

→ it is realistic approach to be used.

Disadvantage of Spiral Model :-

1. Customer Satisfaction :- It is very hard to satisfy the customes that the evolutionary approach will be conducted.

2. Risk Analysis :- This approach requires an risk anslyst or expert to handle the risks properly.

3. Slow Speed :- This Model can't be used so frequently so result are not available soon,

4. High Cost :- It requir a larger sum of money to be completed.

*4. Evolutionary Model :-

→ It is also called Successive version Model or ~~Interact~~ incremental Model.

→ A simple working Model is built.

Application :- large project where you can easily find module for incremental implementation.

Advantages :-

(16)

→ user get a chance to experiment partially developed system.

→ Reduce the error because the core modules get tested

Disadvantage :-

→ It is difficult to divide the problem into several version that would

→

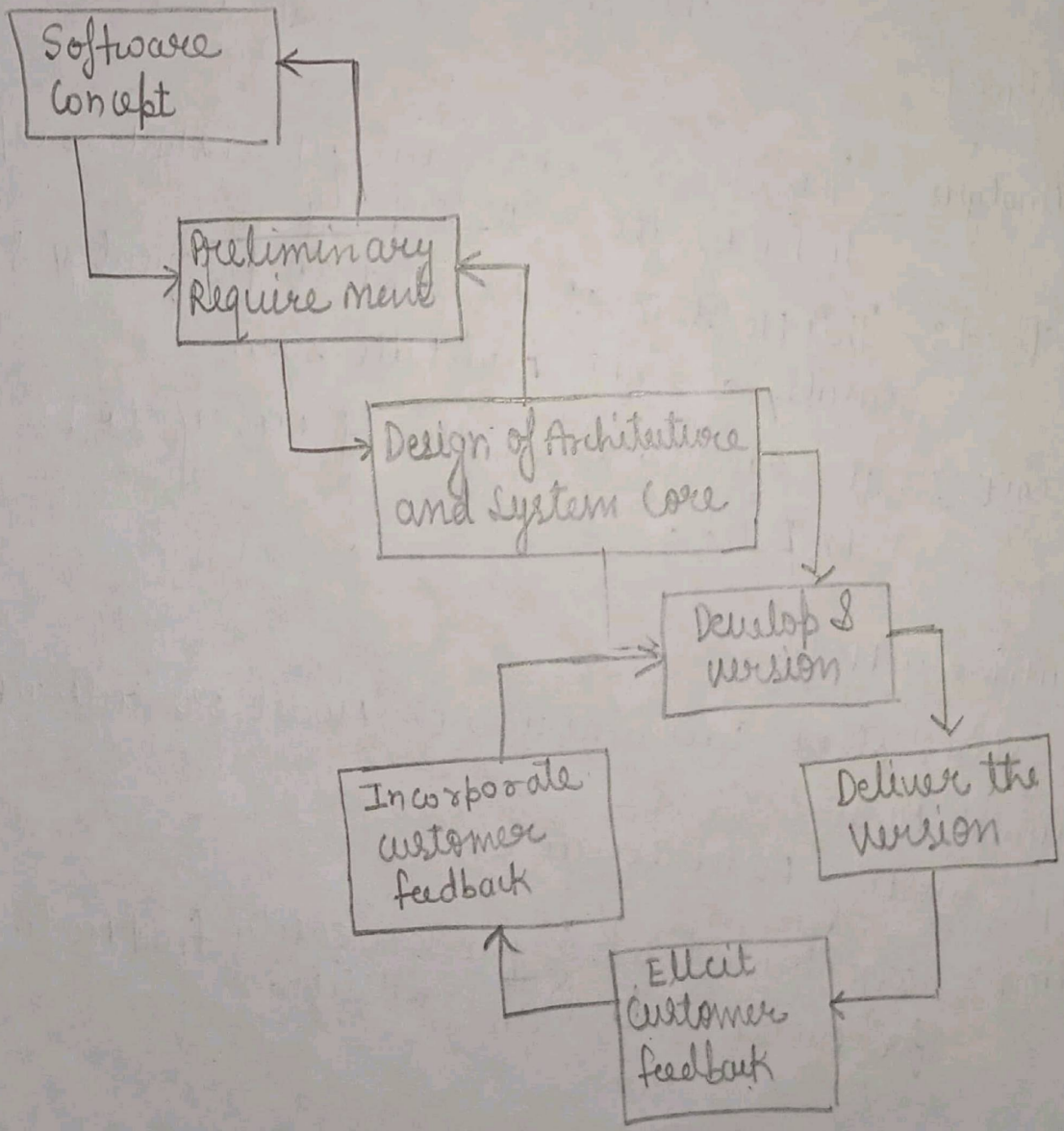


figure :- Evolutionary Model

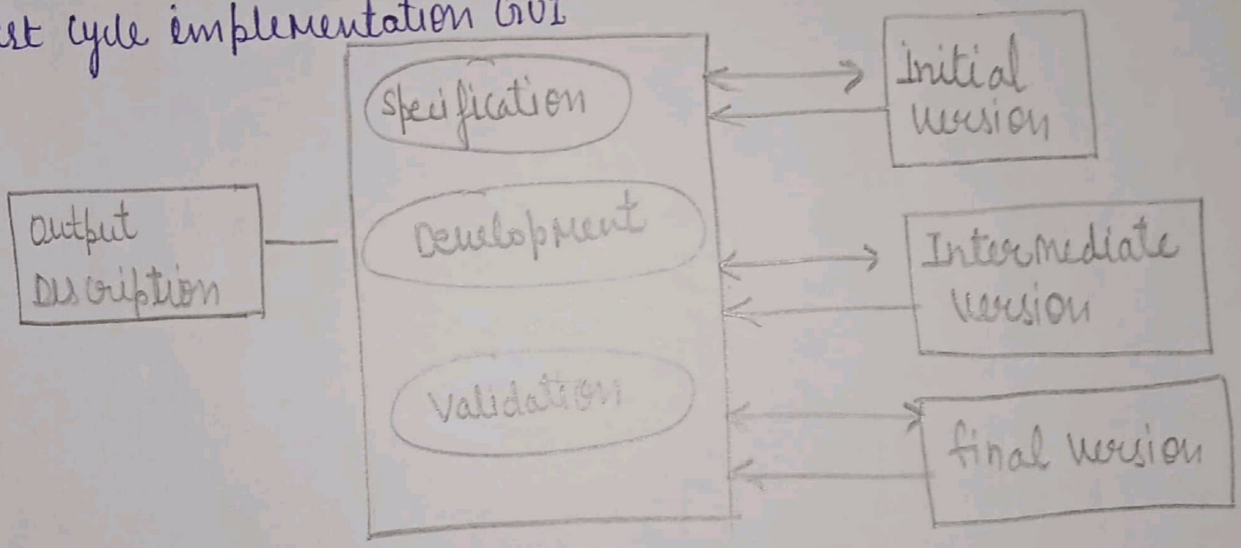
Evolutionary Development Model :-

→ This Model is also known as Evolutionary Model, prototyping Simplated Rapid delivery cycle.

→ This Model have many Similarities with iterative enhancement Model only one and most important difference is that it does not require a usuable product at end of each cycle.

→ For example :-

→ First cycle implementation GUI



* Advantage of Evolutionary development :-

→ Early delivery of partial System.

→ Suitable for project using of new Technology.

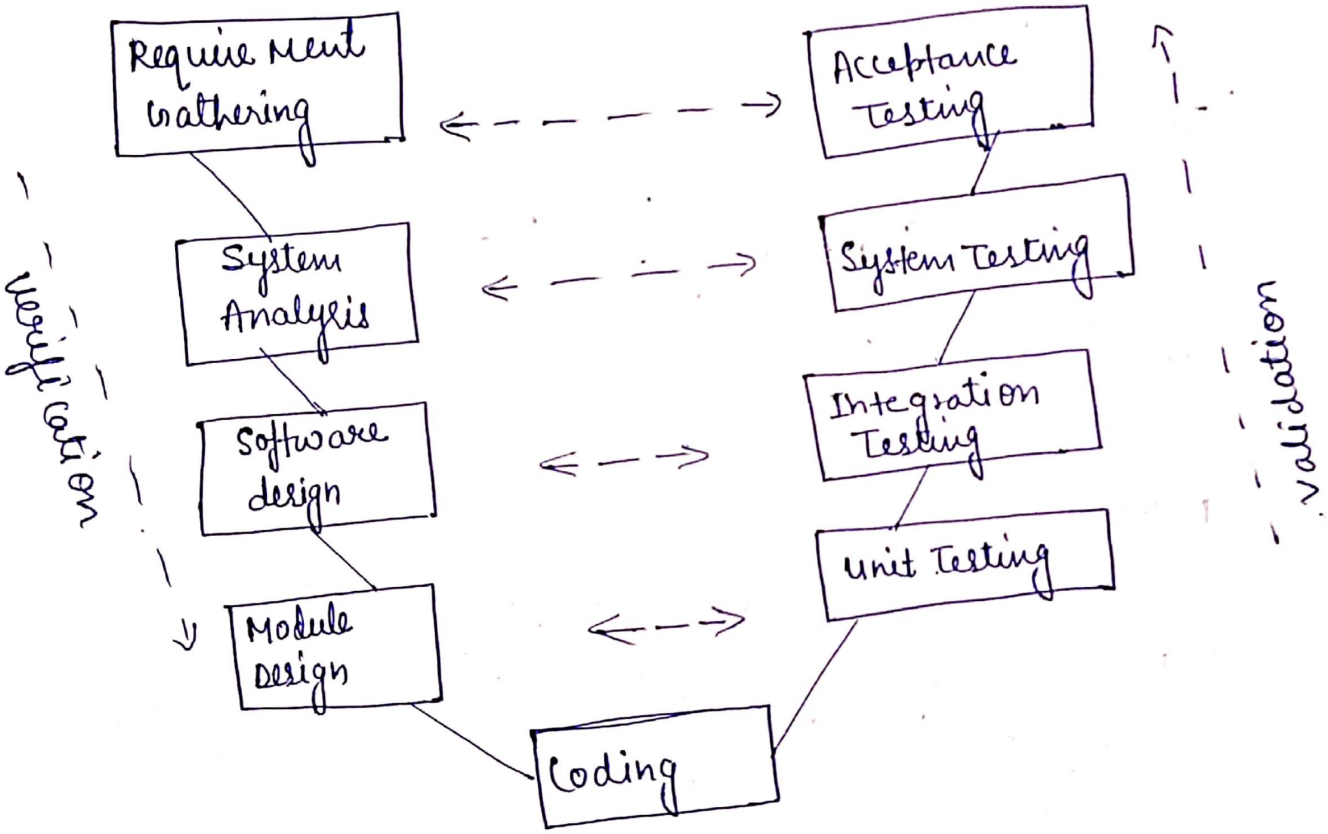
* Disadvantage of Evolutionary Development :-

→ longer elapse time for project if the requirement gathering is done before each increment.

→ Difficulties estimation of cost and Schedule at Start of project.

* V-Model :- The major drawback of waterfall Model is we move to the next stage only when the previous one is finished and there was no chance to go back if something is found wrong in later stages.

→ V-Model provides mean of testing of software at each stage in reverse manner.



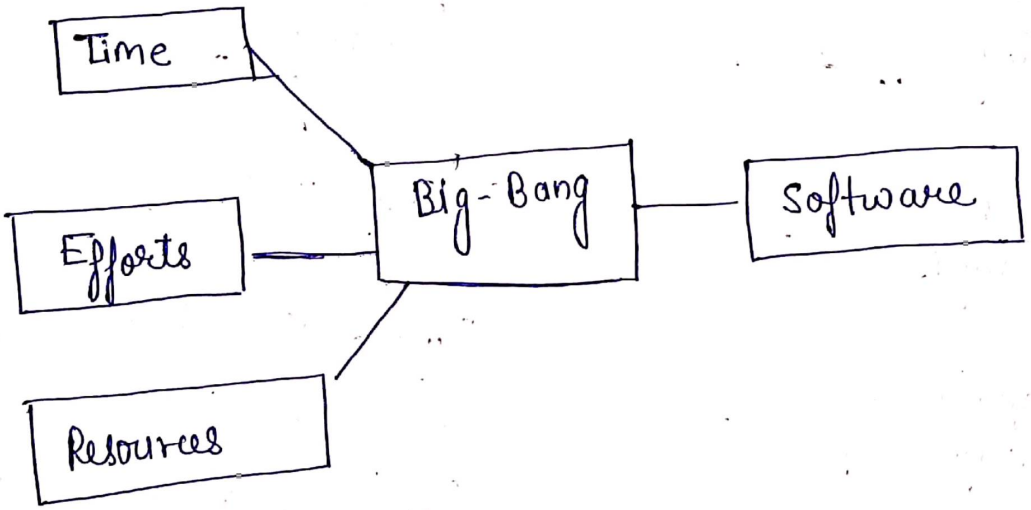
→ At every stage, test plans and test cases are created to verify and validate the product according to the requirement of that stage when the product is developed and is ready for testing, test cases of this stage verify

→ the software against its validity towards when the product is develop and is ready for testing.

*Big Bang Model :- This Model is the simplest Model in its form. It required little planning, lots of programming and lots of funds.

→ This Model is conceptualized around the big bang of universe.

→



→ for this Model, very small amount of planning is required it does not follow any process, or at times the custo -mer is not soure about the requirement and future -needs.

class diagrams :-

⇐ class diagram that represent a set of classes, interfaces and collaborations and

② RAD Model :-

Rapid application development is a software development methodology that focus on building applications in a very short amount of time.

Core Elements of RAD :-

① Business modeling
② Data modeling
③ Process modeling
④ Application generation
⑤ Testing & turnover.

Advantages :-

① Reduced cycle time & improved productivity because multiple team work together parallel.

② Customer is involved throughout the complete cycle minimizes risk of not achieving customer satisfaction and business needs.

Disadvantages :-

(1) RAD require sufficient human resources to create the right no. of RAD team.

(2) RAD is not appropriate where technical risk is high.

When to use RAD Model :-

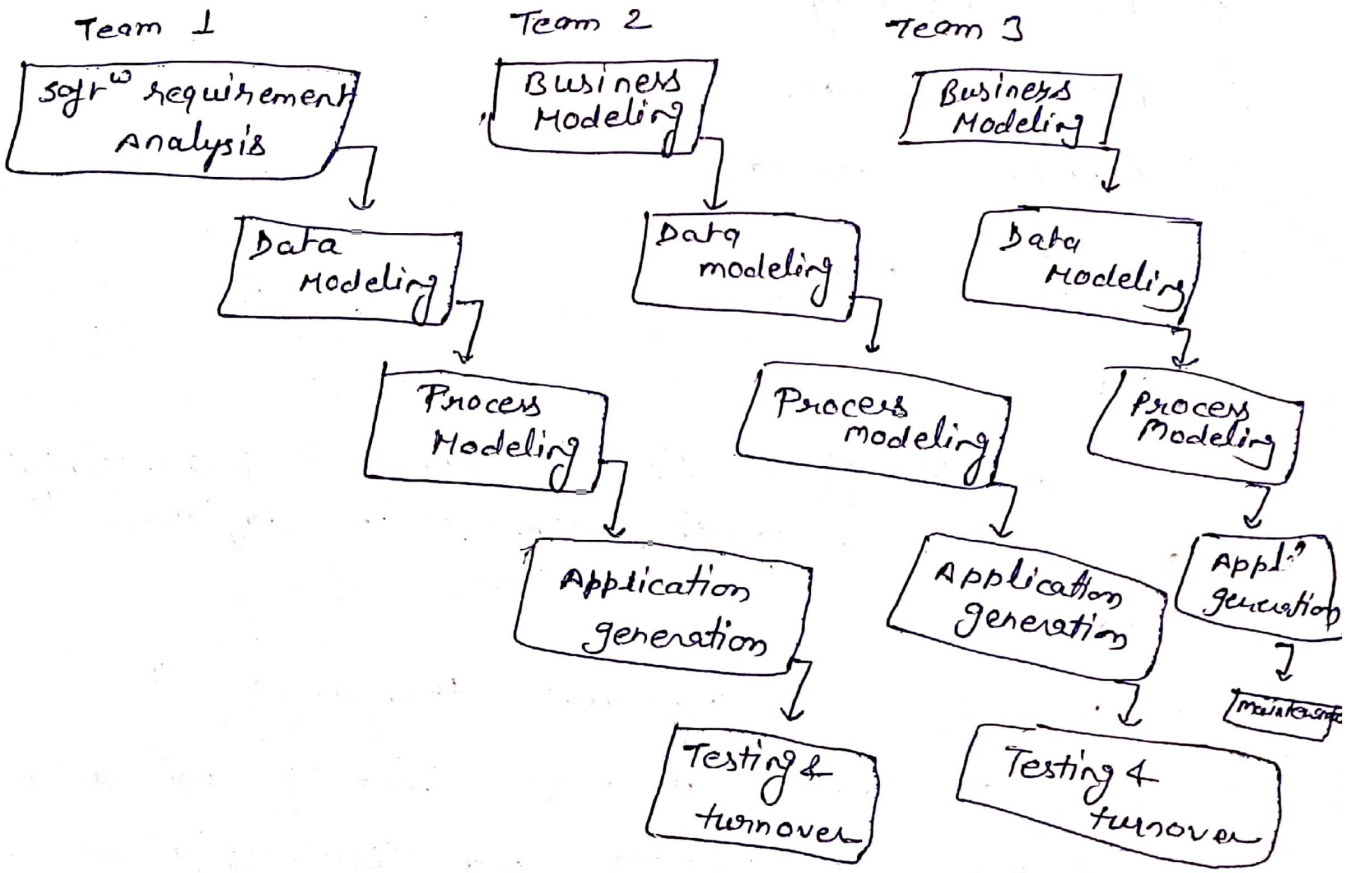
(1) ~~Reasom~~ Reasonably well-Known requirements.

(2) user involved throughout the life cycle.

(3) High performance not required.

(4) Low technical risks.

(5) ~~system~~



Functional Requirement Document

* **Functional Requirement Document (FRD) :-**

→ What is a functional Requirement (FR) ?

→ **Functional Requirement :-** FR is a description of the services that the S/w must offer.

→ it describes a Software System or its component. A function is nothing but inputs to the Software System, its behavior, and outputs.

→ it can be Calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what functional Requirement also called functional Requirement.

* **Types of functional Requirement :-**

* Transaction Handling
* Business Rules
* Certification Requirement
* Reporting Requirement
* Administrative function

Examples of Functional Requirements :-

* The S/w automatically validate computer against the ABC Contract
* The Sales System should allow user to record Customer Sales.
* The S/w System should pass accessability Requirement

* Non functional Requirements :-

→ Non functional Requirements (NFRs) defines System attributes such as Security reliability, performance, maintainability, and on the design of the system across the different

* Types of non functional Requirements :-

* A non functional Requirement defines the performance attributes of a System.
* Scalability Capability
* Availability
* Reliability
* Data integrity etc.

* Examples of non functional Requirements :-

→ The employees never allowed to update their Salary, information
→ Users must change the initially assigned login password immediately after the first successful login.

* Functional versus The user Requirement specification :-

→ The user Requirement

* Verification and validation :-

→ Verification checks whether the software confirms a specification

*Verification :- verification finds the bugs early in the develop-ment cycle.

* Validation :- validation chuks whether the software meets the requirements and expectations.

* Difference between verification and validation :-

	Verification	Validation
1.	The verifying process includes checking documents, design, code, and program.	1. It is a very dynamic mechanism of testing and validating the actual product.
2.	It does not involve executing the code.	2. It always involves executing the code.
3.	Verification uses method like methods link review walkthrough and desk-checking.	3. It uses methods like black box Testing, white box testing and non functional testing.
4.	Verification is done by the QA team while validation.	4. Validation is done by the involvement of testing team with QA team.
5.	Verification process targets on SDLC architecture, design, database etc.	5. Validation process targets the actual SDLC product.

* Verification & validation Techniques :-

There are various techniques used to perform validation and varification of Simulation Model.

* Techniques to perform verification of Simulation Model :-

1) Simulation Model :-

→ by using programming skill to write the debug the program in subprogram.

→ by using "structured walktrough" policy in which more than one person is to read the program.

→ by checking the Simulation Model output the simulation result with various output combinations.

→ by comparing final simulation result with analytic result.

* Techniques to perform validation of Simulation Model :-

*1) Step 1 :- Design a model with high validity it can be achieved using the following steps :-

→ The model must be discussed with the systems experts while designing.

→ The model must interact with the client throughtput the process.

→ output must be supervised of system experts.

Requirement Analysis And Specification

→ Before we start to develop our software, it becomes quite essentials for us to understood and document the exact requirement of the customer.

→ Experience members of the development team carry out this job. They are called a system analyst.

Introduction :-

→ The goal of this phase in to clearly understand the customer requirements and to systematically organise this requirement into a specification document

This phase consists 2 activities. :-

→ 1. Requirement Gathering & analysis
→ 2. Requirement specification.

The engineers who getter and analyze customer Requirement. who and write the SRS are known as system analyst.

* Requirement Gathering & Analysis :-

→ The two main activities involved in this phase are :-

1. Requirement Gathering
2. Analysis

* Requirement Gathering :-

→ Analysis interview end users and customers, study the existing documents to collect all possible information regarding the system.

* Analysis & Gathering Requirements :-

→ The main purpose is to exactly understood the Requirements of the customer certain questions pertaining to the project Project should be clearly understand by the analyst. like

- ✓ what is the problem ?
- ✓ why is it important to solve the problem ?
- ✓ what are the possible solution to the problem ?

* Software Requirement Specification :-

(SRS) Different Categories of users of SRS :-

1. System customer → Specify the require ment and read them to check that they meet theirneeds.
2. Managers → use the Requirement document to Plan a bid for this System and to plan the System development process
3. System Engineering → use the Requirement to understand what System is to be delivered
4. System test Engineer → use the requirement to develop validation test for the Systems
5. System Maintenance Engineers → use the require ment to help understand the System and relation ship between parts.

4. External Interface Requirements :-

① All the possible interactions of the software with people, hardware and other soft^w should be clearly specified.

② The SRS should specify the logical characteristic of each interface b/w the software product and the hardware components for hardware interfacing.

⑤ SRS Representation Guidelines :-

A simple set of guidelines are as follows.

① Representation format and content should be relevent to the problem.

② Information contained within the specification should be nested.

③ Diagrams and other notational forms should be restricted in no. and consistent in use.

(A) Functional Requirements :→

Function requirement specify what output should be produced from the given inputs. So, they basically describe the Connectivity b/w the input and output of the system.

for functional requirement -

(1) A detailed description of all the data input and their sources, must be specified.

(2) All the operations to be performed on the input data to obtain the output should be Specified.

(B) Peyormace Requirements :→

This part of SRS specifies the peyormance Constraints on the software system. These are expressed as "processed transaction per second" or "Response Time from the system for a user event"

(C) Design Constraints :-

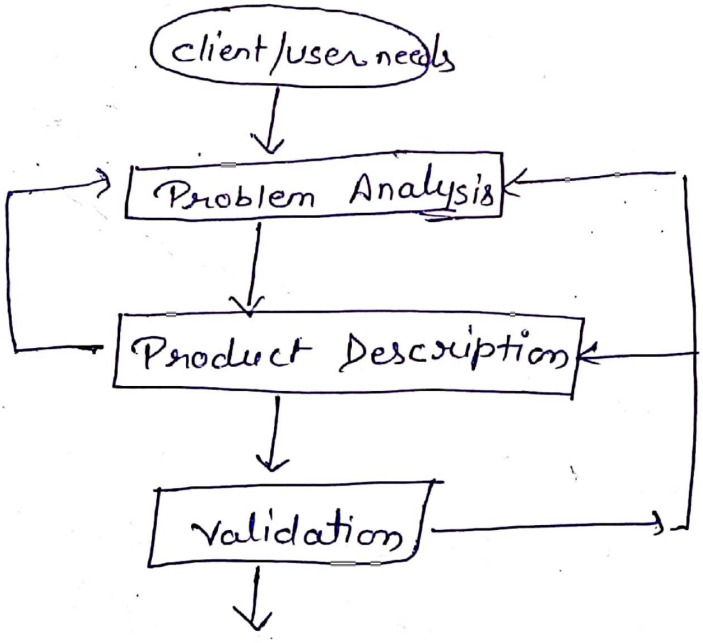
The various design Constraints are standard, Compliance, resource limits, operating environment that may have an impact on the design of the system.

Unit - I

→ SDLC
→ SRS
→ Formal requirement specification
→ Verification and Validation

SRS (Software Requirements Specification) :-

It is a detailed description of a software system to be developed with its functional and non-functional requirements. The SRS is developed based the agreement between Customer and Contractor.



The Requirement Process

① SRS Fundamentals :-

why SRS is required-

① SRS is support needed to maintain the records about all the work done.

② SRS is often a major element to serve a contract b/w client and developer.

③ It specifies the final system or product.

④ It supports maintenance.

⑤

(A) SRS as Communication Tool :→

SRS is a Communication tool because it contains a repository of all work done to date and makes it available to all persons working on large project.

(B) SRS as a Management Tool :-

It gives access to latest work to all project personnel and this reduces the chances of repetition of work.

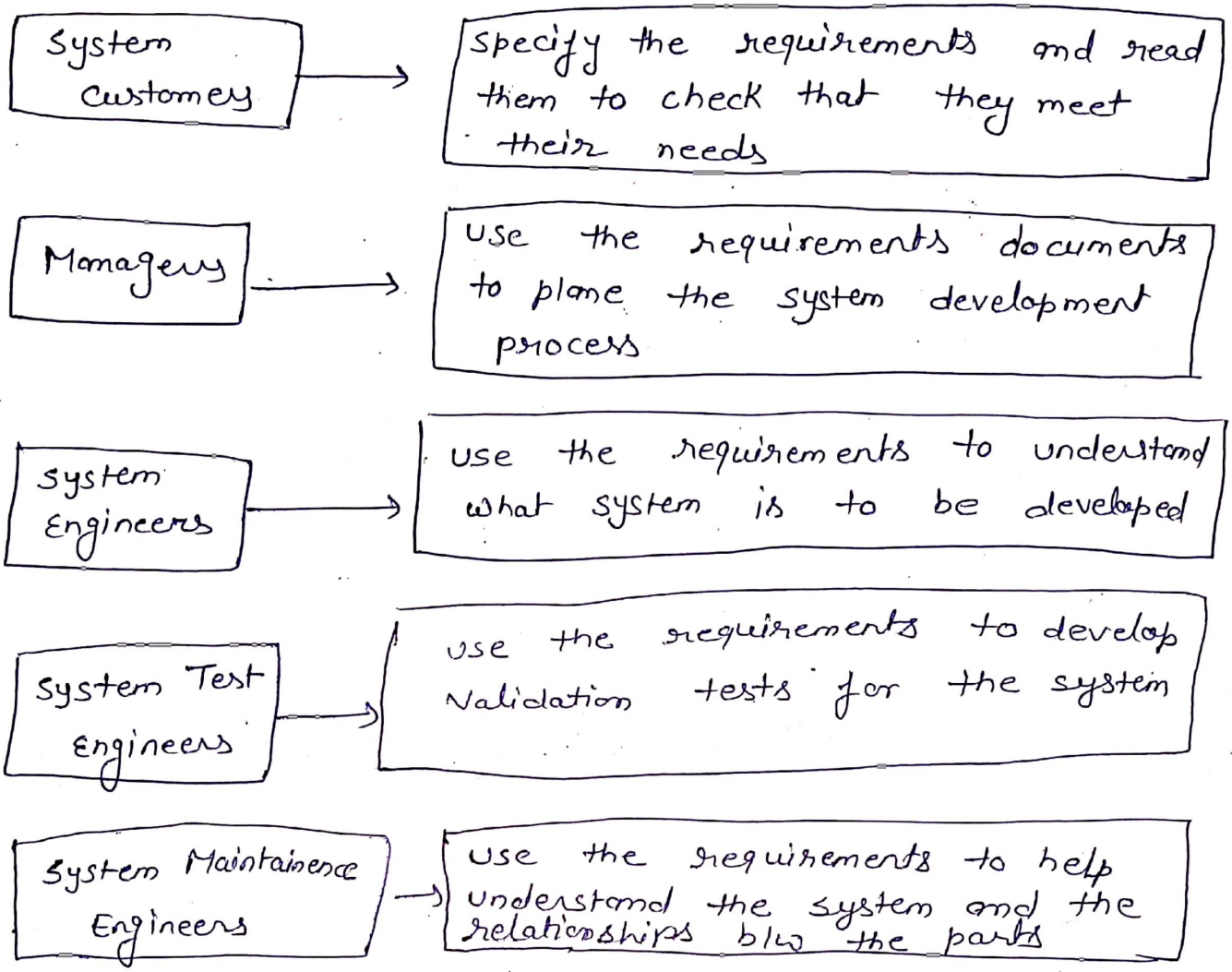
(3) characteristic of the SRS :→

1. Completeness
2. Clarity
3. Correctness.
4. Consistency
5. Verifiability
6. Ranking
7. Modifiability
8. Traceability
9. Feasibility.

(4) Components of the SRS :→

1. Functional Requirements
2. Performance Requirement
3. Design Constraints.
4. External Interface Requirements.

② Users of SRS :→



2.② SRS Principles :→

It is representation process.

① Separate functionality from implementation.
② Define the enviornment in which the system operates : and indicates how ~~a~~
③ ~~Det~~ Develop a model of the desired behaviour of a system.

1) System :-

→ Set of ideas or rules for organizing something, a particular way of doing something.

→ A System is a group of interacting or interrelated entities that form a unified whole.

Difference between System engineering & Software Engineering

System Engineering	Software Engineering
① A System engineer is a person who deals with the overall management of engineering project during life cycle.	① Software Engineering is a person who deals with the designing and developing good quality of software products.
② It deals with logistics, team coordination, automatic Machinery control work process and Similar tools	② Software engineering includes in computer Science or Computer based engineering while System engineering.

Ans-2 Software :- Software is a collection of data or computer instruction that tell the computer how.

→ Software instruction that tell a computer what to do

→ Software comprise the entire set of program, procedure and routines associated, with operation of a Computer System

Advanced Topics in Software Engineering

*1 :- Computer-Aided Slw engineering :- CASE (Software

Computer Aided Software Engineering (CASE) Tool assists Software Engineering Managers and practitioners in every activity associated with the Slw process.

→ CASE Tool can be integrated with in a sophisticated environment.
→ Busines processing engineering tools
→ Project planning tools
→ Risk Analysis tools

* Component-Based Slw Engineering :-

→ Component based slw engineering (CBSE) is a process that emphasizes
→ the design and construction of computer based System using slw. component.

→ Activities :-

1) Component Qualification
2) Component adaption
3) Component Composition
4) Component updates.

(3) Client Server Slw engineering :-

→ There are different slw process Model were ~~such~~ introduced,
→ client Server. System are developed using the classic slw engineering activities Analysis design construction, testing.

Sol(2) Software development :-

Software development is the process of conceiving, specifying designing programming documenting testing and bug fixing involved in creating and maintaining applications, frame works, other Software components.

Software Requirement Specification :-

→ Software Requirement Specification (SRS) :-

→ Software Requirement Specification is a kind of document which is created by a software analyst after the requirement collected from the various sources - the requirement received by the customer written in ordinary language.

→ A software Requirement Specification is a document that captures complete description about how system expected to perform.

Qualities of SRS :-

1. Correct
2. Complete
3. Consistent
4. Verifiable
5. Modifiable
6. Traceable
7.

Soln users of SRS :-

1. System Customer → Specify the requirement and read them to check that they meet their needs

Managers	→	use the requirements document to plane the System development process ③
System Engineers	→	use the requirement to understand what system is to be developed
System Test engineering	→	use the requirement to develop validation test for the system
System Maintenance Engineers	→	use the requirement to help understand the system and the relation ship b/w parts.

List of possible users of SRS

Soln :-

Data Flow diagram :-

DFD is a data flow diagram is a way of representing a flow of data through a process.
DFD maps out of the flow of information for any process or system
It defines symbols input output data, storage point and Symbol like rectangles, circles, and arrows plus short text labels.

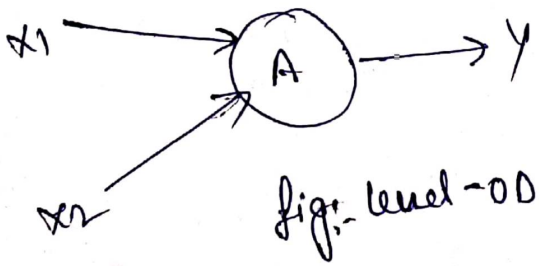
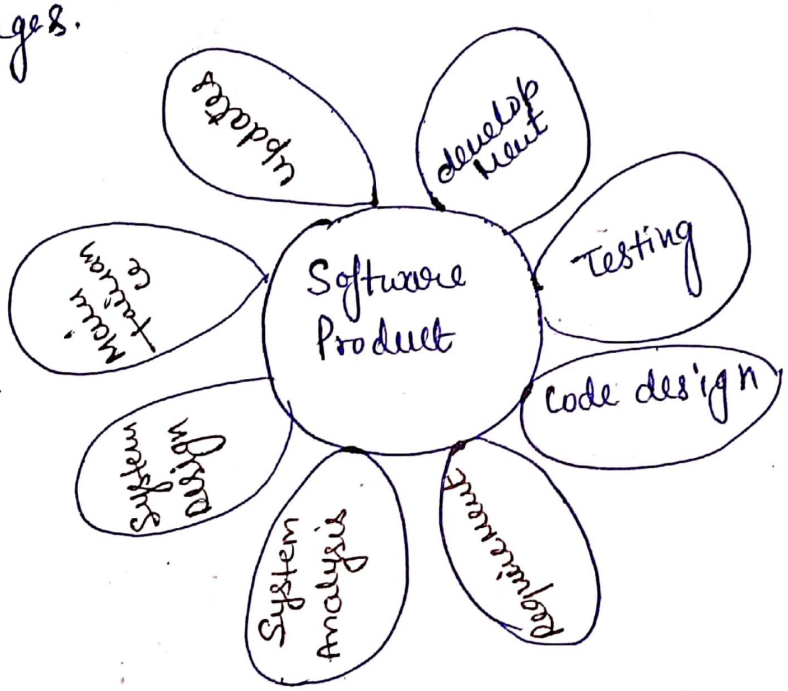


fig:- level-0 DFD

Software :-

Software Engineering :- Software Engineering provides a standard procedure to design and develop a software.

Software :- Software is a collection of integrated programs. Software subset of carefully organized instruction and code written by developers on any of various particular computer languages.



Why is Software Engineering required ?

Soln Software Engineering is required due to the following Reasons :-

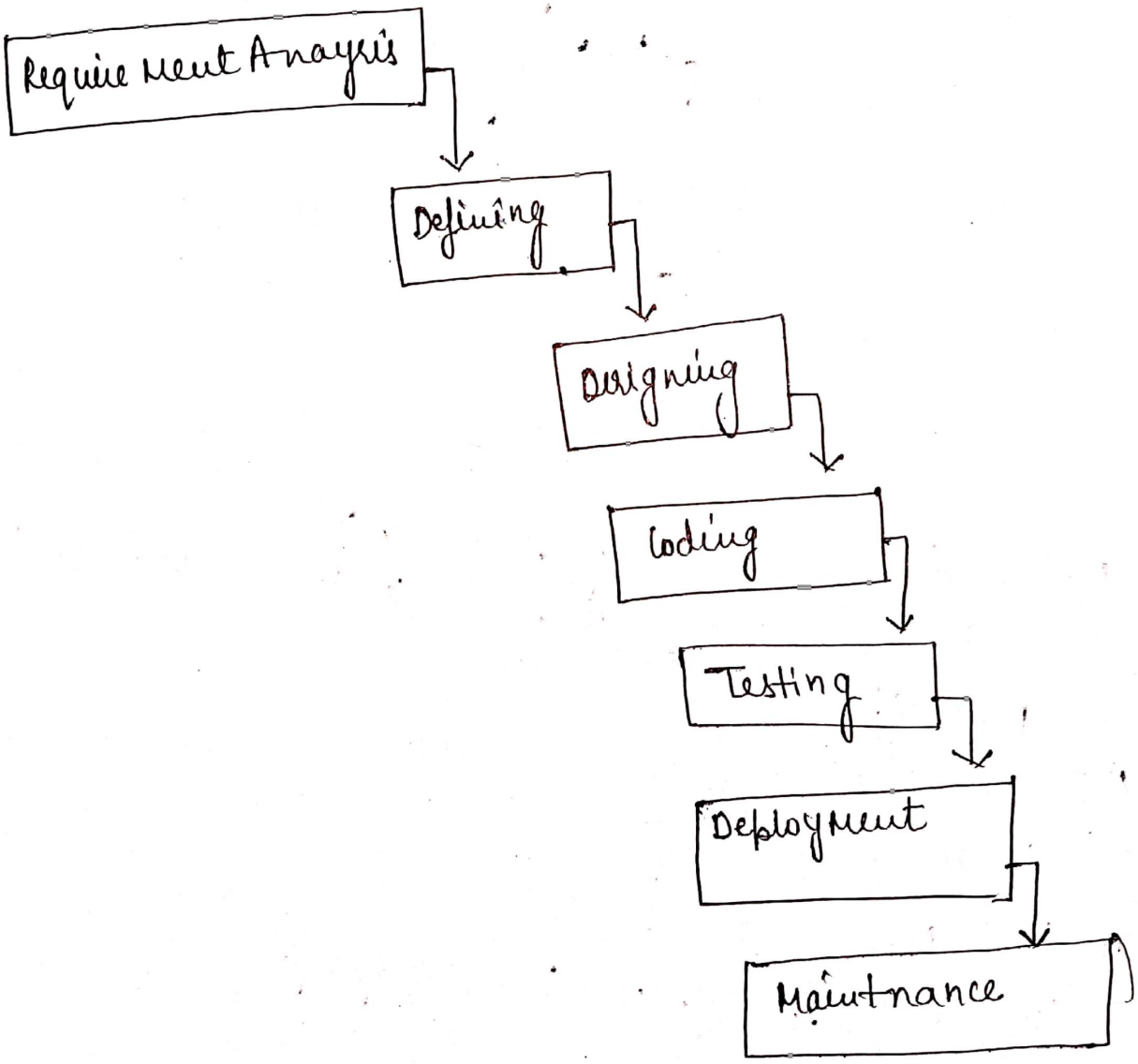
1) To manage Large Software
2) For More Scalability
3) Cost Management
4) To manage the dynamic nature of Software
5) for better quality Management

Soln (2) Describe the Software development life cycle (SDLC)

* Software development life cycle (SWDLC) :-

→ A software life cycle model (also termed process model) is a pictorial and diagrammatic representation of the software life cycle.

A life cycle model represents all the methods required to make a software product transit through its life cycle stages.



Software development life cycle phases.

⑥ Contents of a SRS :→

Recommended structure of SRS

1. Introduction
   - 1.1 purpose of the Requirements document
   - 1.2 Scope of the product
   - 1.3 Definitions, Acronyms or Abbreviation.
   - 1.4 References
   - 1.5 Overview of the Remainder of the document.

2. General description
   - 2.1 Product perspective.
   - 2.2 Product functions
   - 2.3 User characteristics
   - 2.4 General Constraints.
   - 2.5 Assumptions and Dependencies.

3. Specific Requirements.
   - 3.1 ~~specify~~ functional Requirements
     - 3.1.1 Functional Requirement 1
       - 3.1.1.1 Introduction
       - 3.1.1.2 Inputs
       - 3.1.1.3 Processing
       - 3.1.1.4 Outputs.
     - 3.1.2 Functional Requirement 2
     - ⋮
     - 3.1.n Functional Requirement 'n'

3.2 External Interface Requirements
- 3.2.1 user interfaces
- 3.2.2 Hardware interfaces.
- 3.2.3 Software interfaces
- 3.2.4 Communication Interfaces

3.3 Peryormance Requirements

3.4 Design Constraints
- 3.4.1. standard Compliance
- 3.4.2 Hardware Limitations

3.5 Attributes
- 3.5.1 Security
- 3.5.2 Maintainability
- !
- !

3.6 other Requirement
- 3.6.1 Database
- 3.6.2 operations
- 3.6.3 Site Adaption.