

18/11/23.

### Program

- Program is small in size compared to software product.
- No documentation is done.
- Intended for a single user.
- It takes less time.
- It uses less resources.

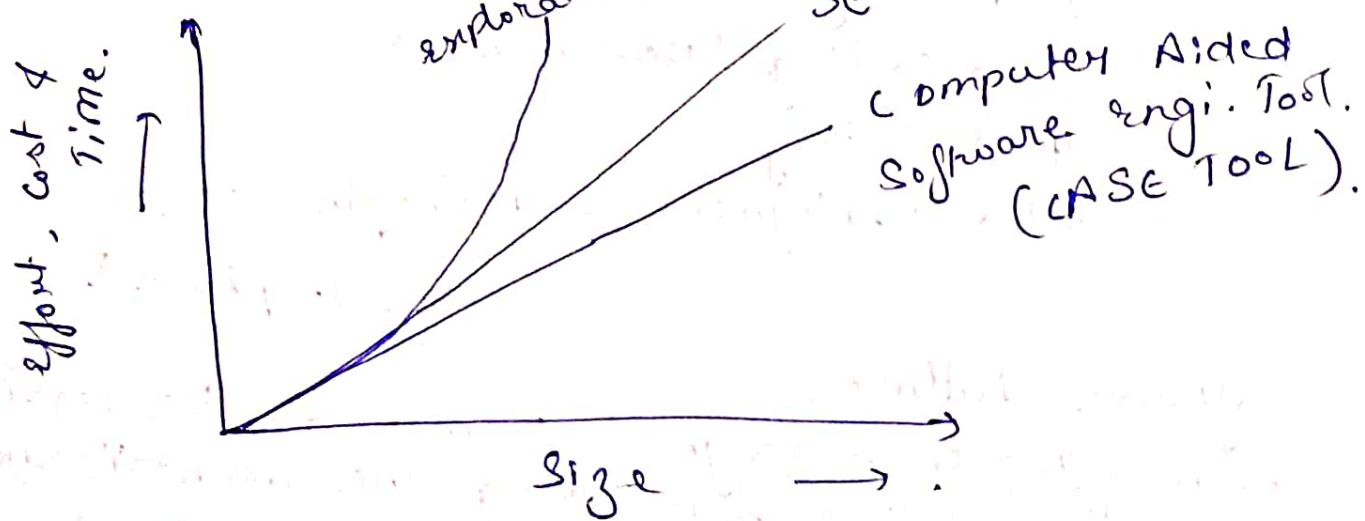
### Product

- Is a group of programs designed for specific task.
- We need proper documentation.
- Intended for a group of users.
- It takes more time.
- It uses more resources.

19/11/23.

### Disadvantages of exploratory style:-

- As the size of project goes increases it compromises
  - ↳ time to build
  - ↳ cost of project.
  - ↳ efforts of understandability increases.



## why SE?

- ↳ given **large software products** → how to develop?
- ↳ how to represent this in a abstract way?
- ↳ How to decompose the large software products into small modules?
- ↳ learn how to handle complex software development.
- ↳ Learning **Team work**.

## Software Life Cycle Models.

- different stages through which an object or product goes.
- similar is with software.
- Phases:
  - ① ↳ Requirement.
  - ② ↳ Design.
  - ③ ↳ Coding / Development.
  - ④ ↳ Testing.
  - ⑤ ↳ Deployment & Maintainence. 60%  
(longest phase)

## Software failure:-

when software is designed to meet all the requirements of users but during the maintenance per phase, the software may face some circumstances where it is unable to handle it then it is software failure.

→ when software is unable to integrate with other softwares.

→ if it is unable to adapt to the new technology or hardware.

### Feasibility Study :-

↳ analysis of whether a project can be taken on not.

↳ cost (financial feasibility).

↳ whether team is capable (technical feasibility).

\* Three stages of study :- (to check for above feasibility).

① Define a problem.

② Find solutions. (if team not sufficient (outsourcing))  
↳ own  
↳ outsourcing ↳ hire new people ↳ Train.

③ Choose Evaluate the solutions & choose the best one.

→ After this feasibility study, the actual phase of project starts i.e. requirement.

→ SRS - Software Requirement System Specifications  
↳ Person involved is system analyst.

↳ Person involved in design - system architect.

① Requirement - output of this phase is SRS.

Stages:-

(1. a) Requirement-

(1. b) Requirement

Gathering + Analysis .  
Specification .

(1.0) Analysis is important so that we gather relevant requirements & remove unwanted & if missing then include.

(1.1) After gathering, there would be a proper documentation of what the customer wants or a user wants. This document is base for other phases.

↳ Specifications includes:-

(i) Functional Requirements :- has input performs processing & gives output. what the model is intended to do.

(ii) Non-functional Requirements :- design layouts; performance;

(iii) Goals of representation :- goal of what we are implementing.

(2) Design :-

↳ Input is the SRS.

↳ Output is design documents.

↳ Imaginative task.

↳ most important as coding is done based on this.

(3) Coding :-

↳ Implementing the design in form of codes.

↳ testing done by development team or coding team is unit testing.

↳ once satisfied, it is sent to the testing.

↳ output source code.

→ Input design documents.

#### ④ Testing :-

- ↳ exhaustive testing is done.
- ↳ explore all possible inputs.

#### Types :-

- (4.a)  $\alpha$  - testing - expected output is known. Development team involved.
- (4.b)  $\beta$  - testing - limited access version is given to customer & when any error occurs then again bugs are fixed.
- (4.c) Acceptance testing.  
↳ Done by other users / clients.

#### ⑤ Deployment & Maintenance :-

- ↳ Test report - input.
- ↳ once deployed only then maintenance is done.
- ↳ before deployment many documents need to be prepared.

#### Types of Maintenance :-

- (5.a) Corrective : - when any bugs are there after deployment then fixes made is called so.
- (5.b) Perfective : - to make existing software perfect on improvements / upgrades.
- (5.c) Adaptive : - to make it adaptive / compatible to new devices / technology.

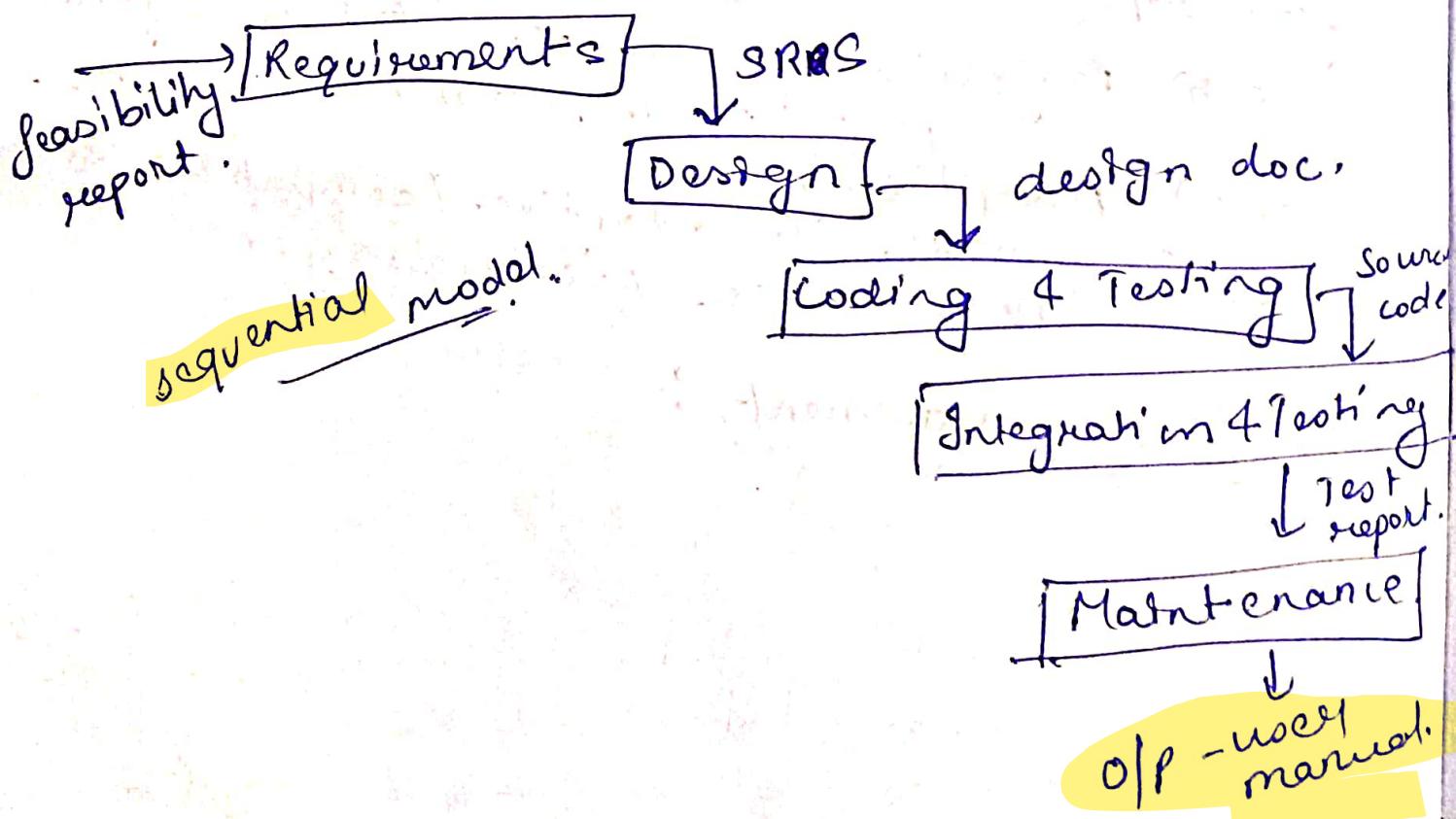
#### Project Management :-

25/1/23. 1/2/23

## Need of Software Development Life Cycle?

- There will be systematic approach for development.
- It will be efficient.
- management team bears all the financial investment for a project.
- 99% Complete Syndrome.
  - ↳ while testing there may be bugs encountered.
  - ↳ The management teams takes report on the project by project manager.
- During testing we may encounter that some requirements are missing. this is 99% complete syndrome.
- To overcome this, phase entry of exit criteria is used.

## SDLC Models — Waterfall Model



→ for every phase we need to clearly define phase entry & phase exit points.

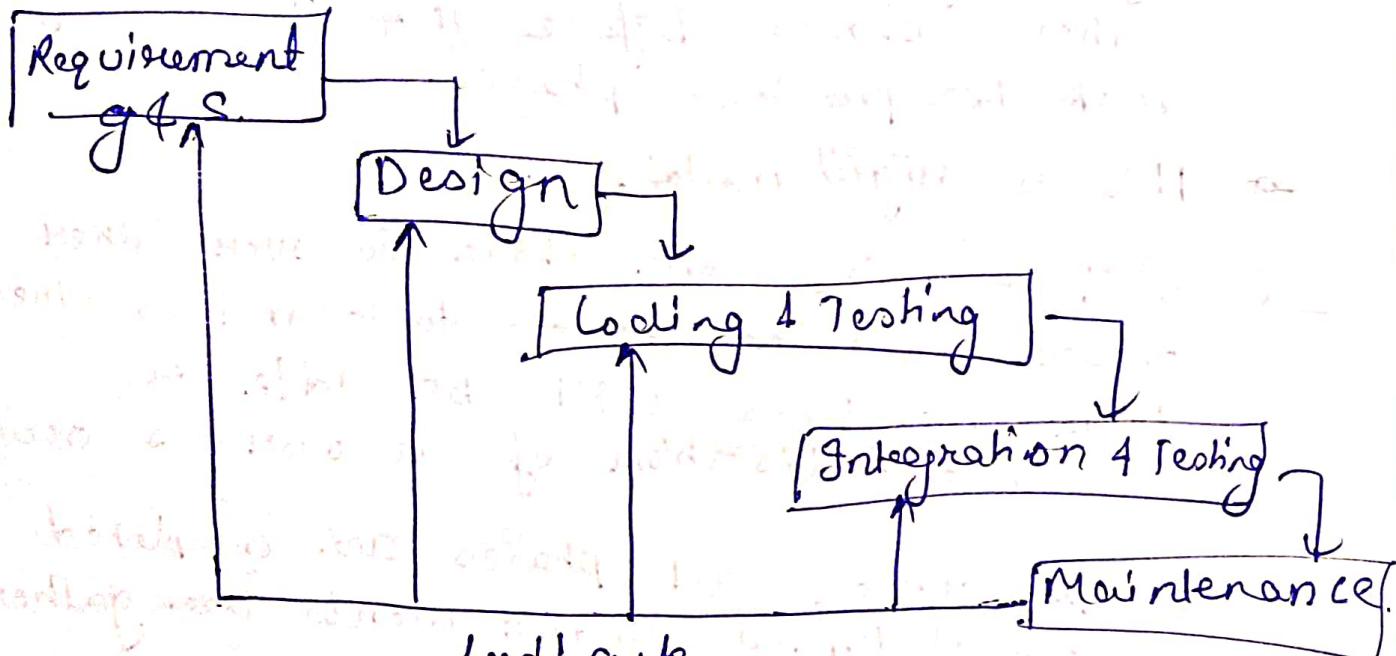
### Drawback:-

- ↳ any changes in any phase affects all the other phases before it & we can't go back to previous phases.
- This is rigid model.
- until & unless one phase is over other phases will not start and so other team members will be idle. So underutilisation of resources occurs.
- It presumes all phases are completed successfully or requirements are gathered completely for the development.
- risk modelling is not handled. (i.e. if some exceptions occur).

- 
- This model is used for preparing software document & also forms based for other models.
  - Software Document is needed because if some other person comes or is appointed for maintenance, then he must know for what all things were used in development.

## Incremental waterfall Model:

- same as waterfall model but additional feature of giving feedback to previous stages.



### Drawbacks:-

- time consuming & costly.
- wastage of resources.
- risk modeling not handled.

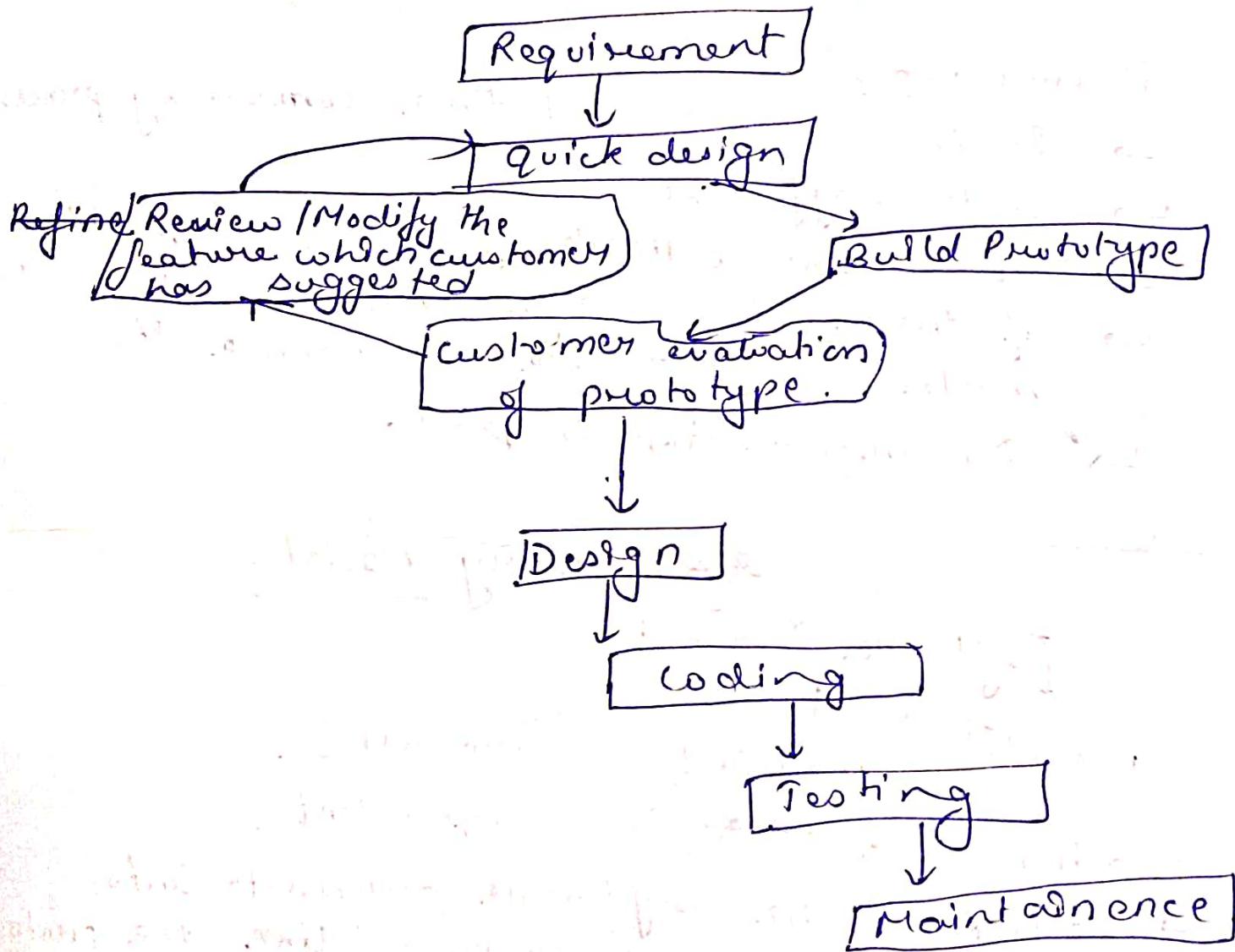
### Uses:-

- used for developing small projects.
- " " well known projects.
- Developing project where we have experienced team.

2/21/23

## Prototype Model :-

- prototype is a toy implementation of the actual software.
- where can we use this model?
  - ↳ this model can be employed where there is an absence of detailed information regarding the input to the system, processing needs & output requirements.
- it is a software development model in which prototype is built, tested and reworked until an acceptable prototype is achieved.



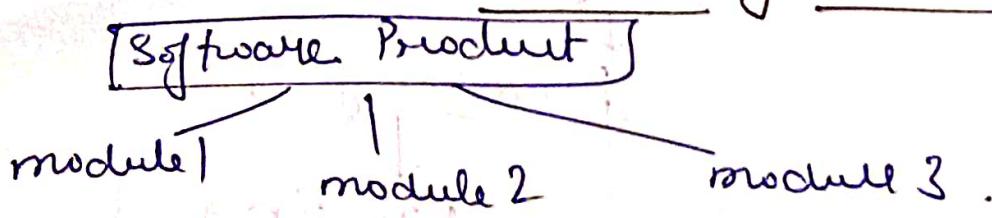
### Advantages:-

- ↳ efficient compared to actual model.
- ↳ reduce the risk:- missing functionality can be identified, which helps to reduce the risk of failure.
- ↳ Identifies the risk:- users are actively involved in development. Therefore, errors can be detected in initial stage of S.D. process.
  - Better user experience (customer satisfaction ends).

### Drawbacks:-

- ↳ It is a slow and time consuming process.
- ↳ Cost ineffective.
- ↳ It is not suitable for complex problem.
- ↳ If there is a large risk then this model is not able to handle it.
- ↳ Documentation is poor.

### Evolutionary Model



- Each module ~~are~~ is independent.
- decompose the software product into modules or smaller versions (like the photo type model) and add one feature of taking feedback like incremental waterfall model.

- If this subpart is given to the customer then -
- changes can be done easily.
  - bugs can be fixed easily.
  - customers don't have to wait for completion of project.

→ Here the software is given into parts to the customer -

(A)



(A)\*

added few more things



full product.

A initially made

→ There will be no load on the development team to complete it fully and also customer need not to wait.

→ "Design a little, build a little, test a little, deploy a little."

Requirements

↓  
Identify the core modules.

↓  
Develop the core module.

↓  
Collect feedback from customer

↓  
Develop the features using TWF M.  
Once all features are developed.

Deployment of Maintainence.

## Advantages:-

- no financial burden on either side.
- changes can be done easily.
- bugs can be fixed.
- customers don't need to wait for completion of project.
- it make feature robust.

## Disadvantages:-

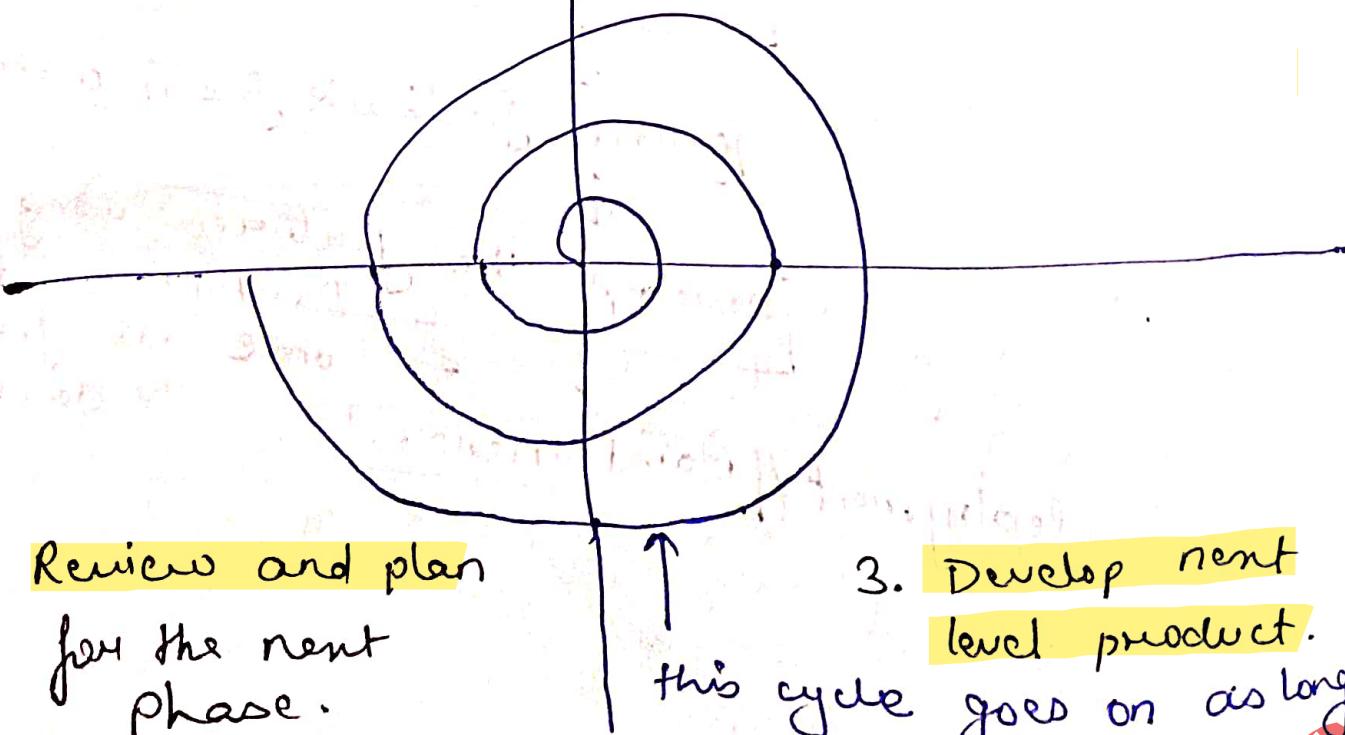
- it take more time.
- all problems / product may not be decomposed into core modules.

## Spiral Model :-

(Meta Model)

1. Determine objectives  
+ identify alternative solutions.

2. Identify & resolve risks.



4. Review and plan for the next phase.

3. Develop next level product.

this cycle goes on as long as the manager is not satisfied.

## steps:-

- ① Requirements are gathered from customers and objectives are identified. Alternate solutions for the phases are proposed.
- ② all the possible solutions are evaluated to select the best possible solution. Then the risks associated with that solution are identified and are resolved. At end, the prototype is built for best possible solution.
- ③ Identified features are developed and verified through testing.
- ④ The customers evaluate the so far developed version of software. In end planning of next phase is started based on the evaluation by customers (step 1).

→ It incorporates all the previous models and hence is called meta model.

## Advantages:-

- Risk handling is done properly.
- It is used for developing a complex problem.
- customer satisfaction.

## Pis advantages:-

- It is time consuming
- expensive
- It needs experienced people.

8/2/23

- feasibility report.
- functional requirements.

## # Phase Containment Error.

Identify the phase error is that phase only and not after its completion.

15/2/23

- System :- any entity that changes it's states upon giving any input.

→ to check whether SRS<sup>1</sup> is correct or not or complete we use formal specification.

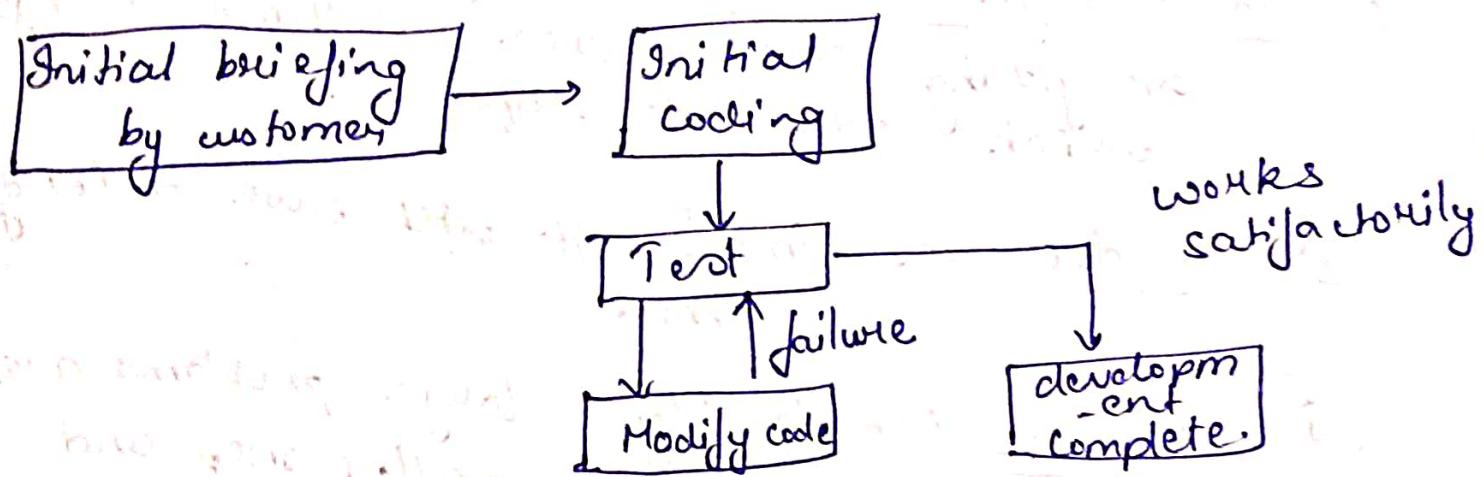
→ Side effects :-

\* \* .

\*. Software Crisis :- The term was used in early days of comp. science for difficulty in writing useful programs in required time. This was due to rapid increase in computer power and software complexity, that could not be tackled. With increase in complexity of software, many problems arose because existing methods were inadequate to solve it (like the exploratory style).

## Exploratory Style of development :-

- It also refers to the informal development style or build & fix style in which programmer uses his own intuition to develop a program rather than using a systematic approach towards solving & developing.
- This is quickly developed and bugs are fixed whenever it arises.



## Disadvantages

- With increase in size of project the development time, development cost and efforts increases exponentially.
- It leads to unmaintainable code, program mining without planning leads to the to unstructured and poor quality of code.
- It is difficult style when there is a proper developing team because in this style every developer uses his own intuition to build.

Software engineering:- SSD approach

→ It is a systematic, scientific and disciplined approach to the development, functioning and maintenance of software.

### Need for SE?

- ① Handling big projects:- to avoid the problem of software crisis.
- ② To manage the cost- programmers plan everything and reduce all those things which are not required.
- ③ To decrease time- It will save a lot of time.
- ④ Reduces complexity:- large problems are broken down into smaller ones and solved one at a time in software eng.