SOFTWARE

It is a collection of techniques, methodologies & tools that helps with the production of:

1) High quality software
2) Before the given deadline
3) Within a given budget
*) With circulating changes

Difference Computer Sc. & Software Engineering

→ CS is concerned with fundamentals & theories.

→ SE is related with practicality of developing and delivering useful softwares.

Difference b/w System Engineering & Software Engg.

→ It is concerned with all aspects of computer based system development. It includes system specification, system architecture & system design for software, hardware & process.

→ It is a part of system design concerned with developing softw -are infrastructure, control app -lications & databases.

SOFTWARE PROCESS MODEL

1) Specification:
   → What the software will do.
   → Development constraints : Designing, Coding
   → Validation : Checking the functionality if it is what the user wants.
   → Evolution : changing the software as per change is demand

SOFTWARE DEVELOPMENT LIFE - CYCLE (SDLC)

1. Requirement Gathering / Communication
2. Feaseability study (finding out limitations or what we can do & what we cannot)
   a. Time
   b. Resources
   c. Program
3. System Design (Creates blueprint for system)
   System representation by looks & feel.
4. Coding
   (Returns our product)
5. Testing
6. Maintainance

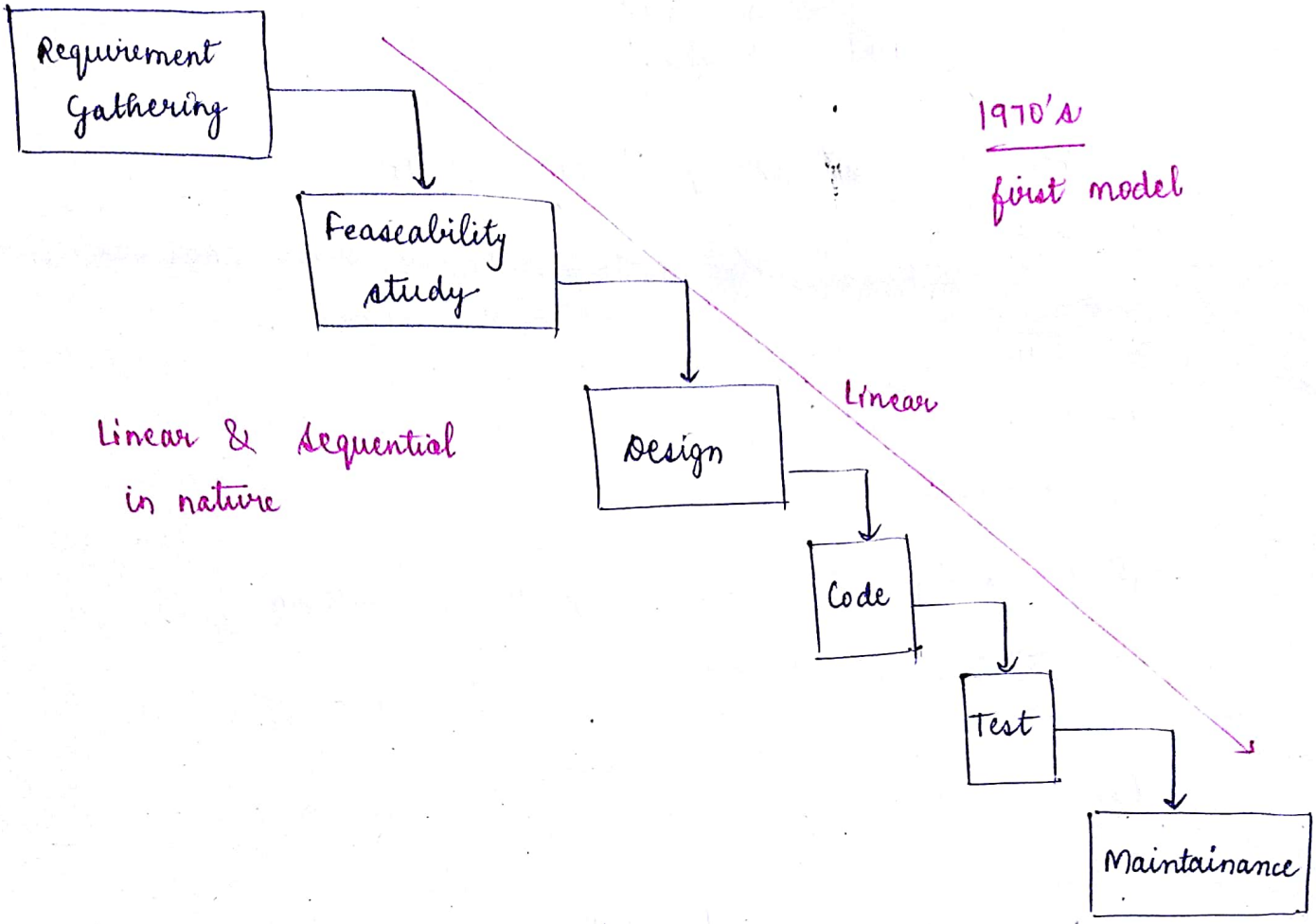
Feedback : Response of user after using the product

MODEL STUDY

1. Waterfall Model

* Property : Cannot revert it back , continuously flowing to the next steps

* Diagram :



1970's
first model

Linear & Sequential in nature

① Each phase must be fully completed before going to the next stage because back stage movement is not allowed.
② There is no overlapping b/w the phase.

Advantages of Waterfall Model :

① Simple to understand & implement both.
② This is appropriate for smaller projects where requirements are less.
③ They are easy to manage.

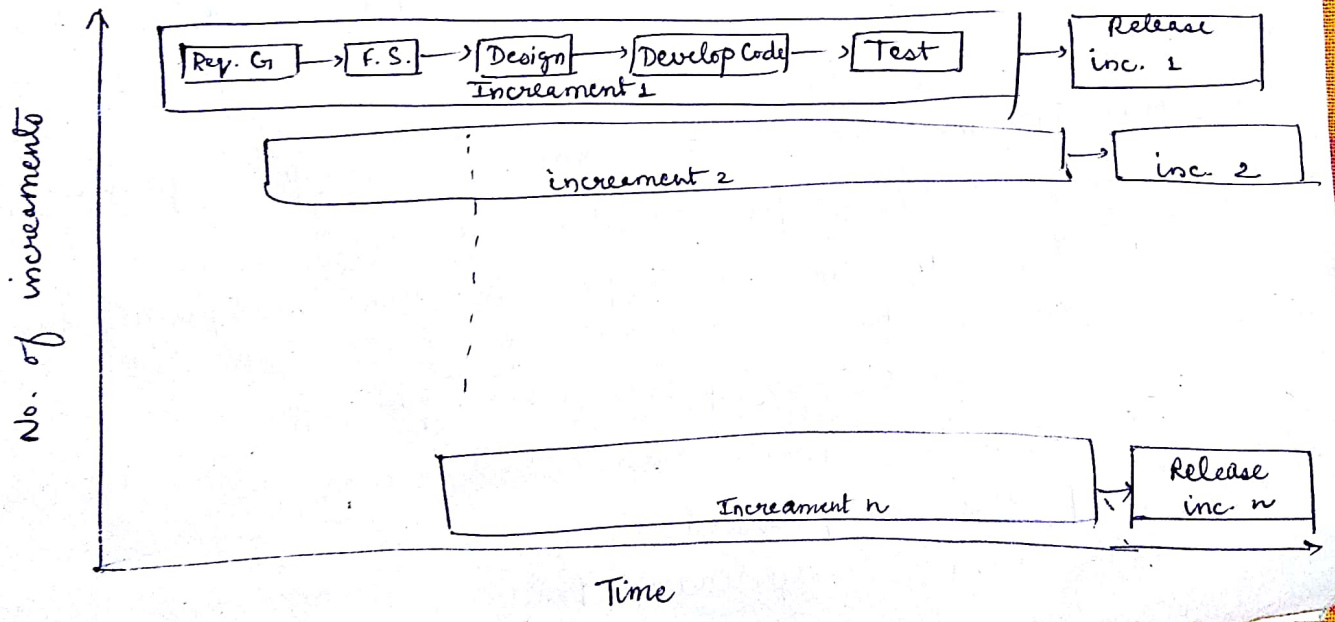
* Extended Waterfall Model to be studied on your own.

Disadvantages of Waterfall Model :

1. Cannot be used for larger projects.
2. No user interaction in the whole process.
3. No back tracking is possible
4. High risk factor
5. Not appropriate when the technology is changing fast.

2) Increamental / Iterative Model :

1. This model does not start with full specification & runs under multiple development cycles - ~~It~~ each iteration passes through requirement, design, implement & test.
2. Each increament deliver a part of required functionality
3. Development is broken into no. of increaments



Advantages of Increamented Model

1. Less time will be requirement.
2. Each increament will work as a prototype.
3. Easy to debug.
4. Risk is lesser than waterfall model.

Disadvantages

1. No overlapping b/w the phases (no comm.)
2. If the system is not modular than this can't be applied.
3. ~~Used for~~ Good for larger products (time decreases cost too decrease)

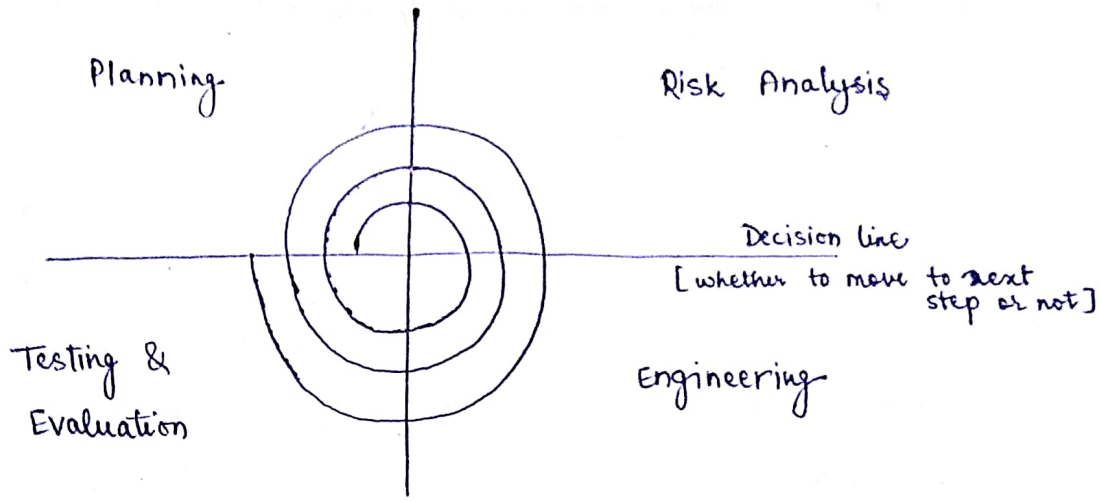
16/8/18

3. Spiral Model :

1) It was presented by the person called & the majority of this process depends on risk management.
2) It has the feature of prototype & incremental model.
3) It has 4 phases :
   (a) Planning Phase : It consist of requirement gathering & identifying probable solutions.
   (b) Risk Analysis : Identifying the risk in applying the sol^n & finding alternate solutions if present. This phase also produces a prototype.
   (c) Engineering Phase : This is the phase where the actual product is produced.
   (d) Testing & evaluation Phase : Software is tested with

multiple used cases that arrives in the software.

EVALUATION : The costumer checks that the software is behaving as per the expectations.



Advantages of Spiral Model :

1. High risk Analysis
2. Early detection of error & flaws.
3. Good for mission critical & large projects
4. Software is produced in early stage.

Disadvantages :

1. It is a costly model (requires cost for risk analysis)
2. High specific techniques are required.
3. Not useful for small projects.

4) RAD : Rapid Application (Development) Model

① It makes use of reusable software components.
② It is a high speed adaption of linear sequential model.
③ It generates an emphasis on component based construction & has following phases.

ⓐ Business Modeling : In this the overall architecture of the required software, available components impact on cost & execution is defined.

ⓑ Data Modeling : How the data of each module will interact with each other & generates required results will lie under data modeling.

ⓒ Process Modeling : This defines the execution strategy of different modules & interactions between them.

ⓓ Application Generation : This deals with generating the actual product with the integration of reused & self designed components.

ⓔ Testing : To check whether all the components are working in synchronisation in all the desired o/p will be produced.

Advantages of RAD

① It is faster then other models.
② It is cost efficient.
③ High performance because reused components are tested &

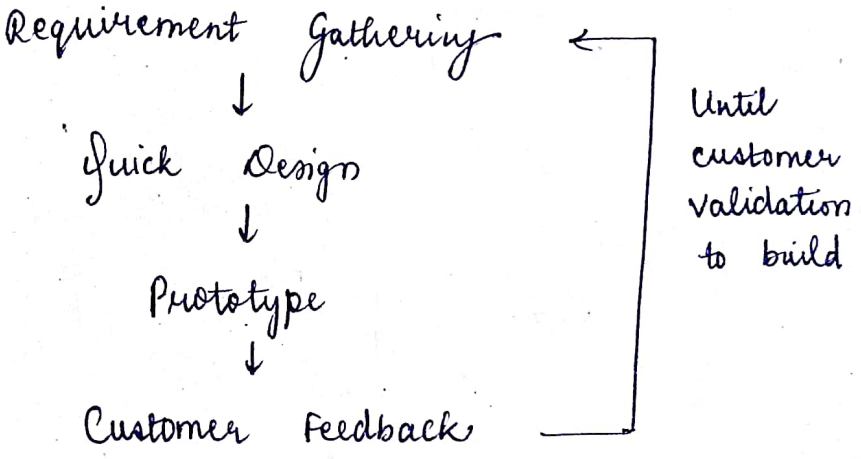
already in use.

Disadvantages :

① Integration of the components can be messy.
② It would be costly when the components in use are taking larger then expected in producing desired o/p.

5) Prototype Model :

In this a prototype is built, tested & rework as necessary until the exceptable prototype is finally achieved from which the actual system can be built



Advantages :

① Proper understanding of user requirement.
② Design feasability is easy.

Disadvantages :

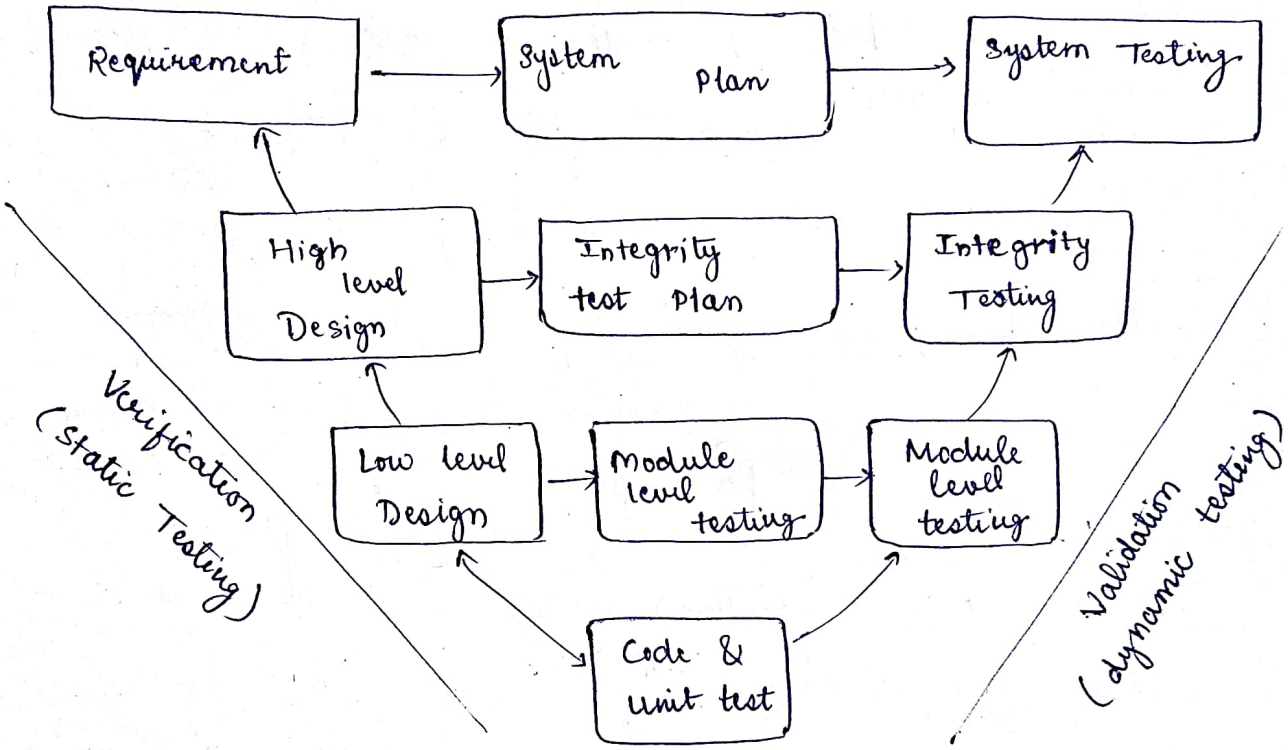
① It is time consuming.
② User treats prototype as it solutions

(3) This produces only partial solution. 17/8/18

6.) Validation Model :

→ It is a variant of waterfall model.

→ It associates each activity with validation at some level of abstraction. Each development model builds activity of the system & with the next level put abstraction on it after applying validation.



Advantages :

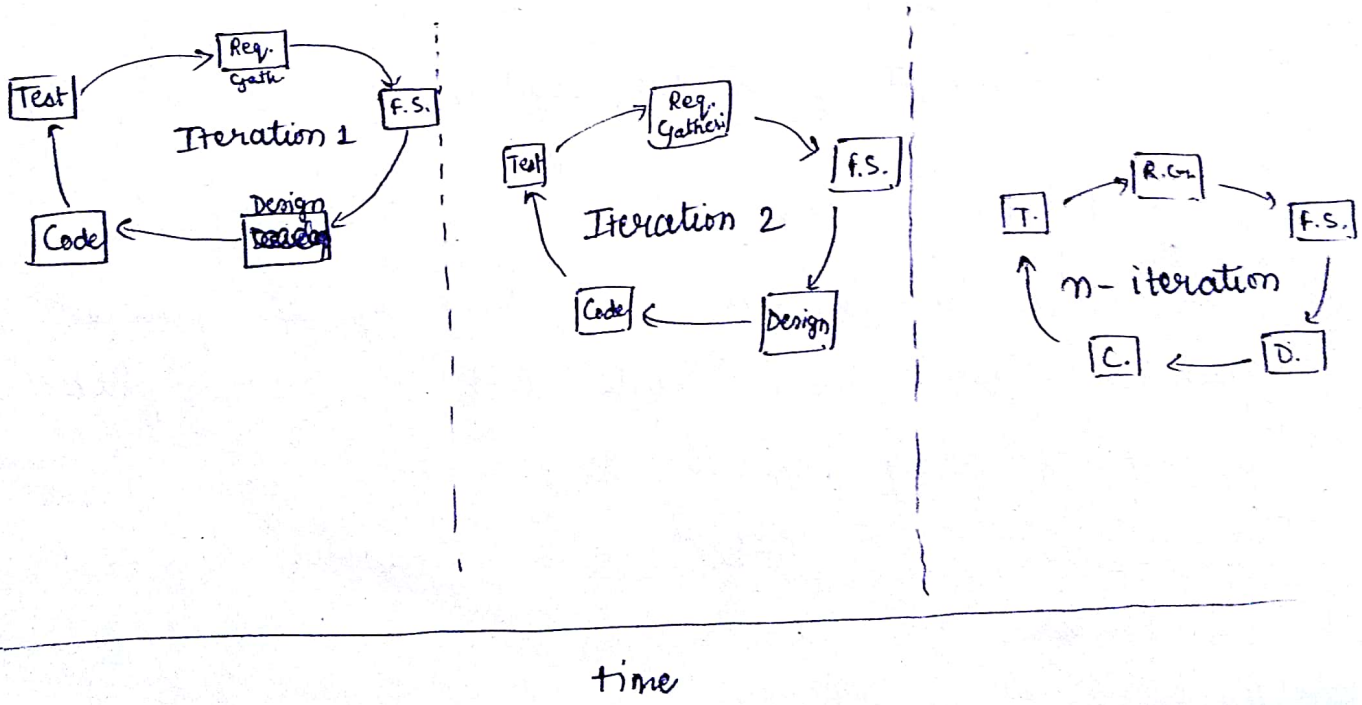
(1) Simple & easy to use
(2) Each phase has a specific delivery
(3) Higher chances of success
(4) Works well for larger projects

Disadvantages

1. Very rigid in nature as well as expensive in nature (Testing requires experts).
2. Early prototypes are produced.
3. Not a clear path is provided how problems will be solved at each phase.

7.) Agile Model :

1) In this all the task are divided into time boxes (small time-frames) to deliver specific release within specific time with current technologies. In this iterative approach is taken & working model is built with each iteration



Advantages of Agile Model:

1) It consists of multiple interaction with user.
2) A software is produced at each stage of iteration hence changes can be easily colaberated.
3) Customer colaboration is high, hence higher chances to meet the expectations.
4) Very responsive towards the changes.

Disadvantages:

1) Not suitable when the dependencies are complex & it cannot be made sure what will be produced at each iteration.
2) There is a risk for making extensions in already existing system.
3) Higher dependancy on the customer.

8.) Big Bank Model

1) This is the model where not any specific process is followed. Development starts with very little planning & understanding of the requirements.
2) Development starts with available resources like money & efforts.
3) Requirements are build & understood as they appear in the system.

Advantages:

1) It is a good learning aid for students.
2) It is easy to manage.
3) Could be started with little resources.
4) Very cheap & simple.

Disadvantages:

1) High risk, high uncertainity
2) Very poor model for ongoing projects.
3) It becomes very expensive when requirements ~~Are~~ are misunderstood.

SRS : Software Requirement Specification

→ It is an official statement of what the system developers should implement. It is a complete description & behaviour of the system.

→ It has used requirements & system specifications.

→ Apart from requirements all the performance measures are defined under this.

Characteristics of a good SRS :

① CORRECT: Every requirement should be listed in SRS.

② UNAMBIQUS : Every requirement should have only one interpretation.
(when 2 things look similar)

↳ Project perspective : It provides further the product to

3. COMPLETE : It should contain all the hardware, software & performance requirements.

4. CONSISTENT : Singular terms are used at each places.

5. RANK & IMPORTANCE : It should be properly divided into essential v/s desirable.

6. VERIFIABLE : It should be finite & could be properly tested.

7. MODIFIABLE : Permits effective modification.

8. TRACABLE : Original requirements could easily be mapped with development in real life.

Structure of SRS :

1) Introduction :

↳ Purpose : This defines why the software is developed & intended audience for SRS.

↳ Scope : The benefits & objectives of SRS. It says what the software will & will not do.

↳ Definations : It provides the defination of the terms used & listed abbreviations.

↳ References : List of all the documents that are used to develop the system.

↳ Overview : It provides brief description of rest of the SRS. & how it is arranged.

2) Overall Description :

↳ Project perspective : It provides whether the product is dependent or self contained & briefs the function of each component.

↳ Project Functions : It has summary of the functions that the software will perform & block diagram of different functions & their relationships.

↳ User Characteristics : The general characteristics of the user that will affect the system.

↳ Constraints : General description of items & limited developer options.

↳ Assumptions & Dependencies : The assumptions that have taken during object finalization & component construction

3.) Specific Requirements :

↳ External Interface & Requirements : This provides characteristics that a software most have to support human interface with the system.
Eg : Diagram of screens & page layout content.

↳ Performance Requirements : Capacity of the system, Response time of the system & system priorities with decide the performance requirements.

↳ Design Constraints : Company policy, Hardware limitations

Formal Specification

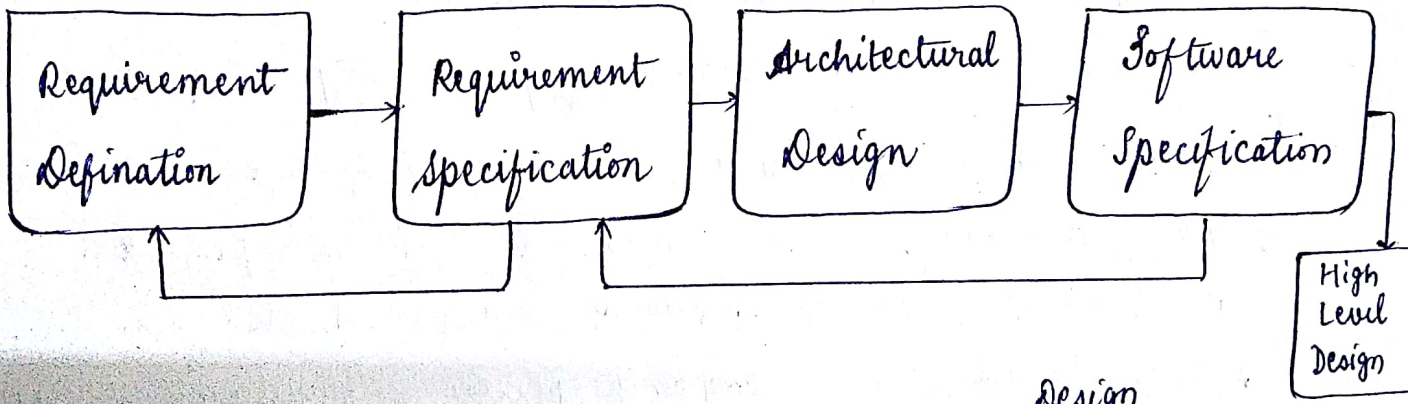
These are the techniques required for unambiguous (not simple) specificati of software. This helps to discover the problems in system. This describes the use of:

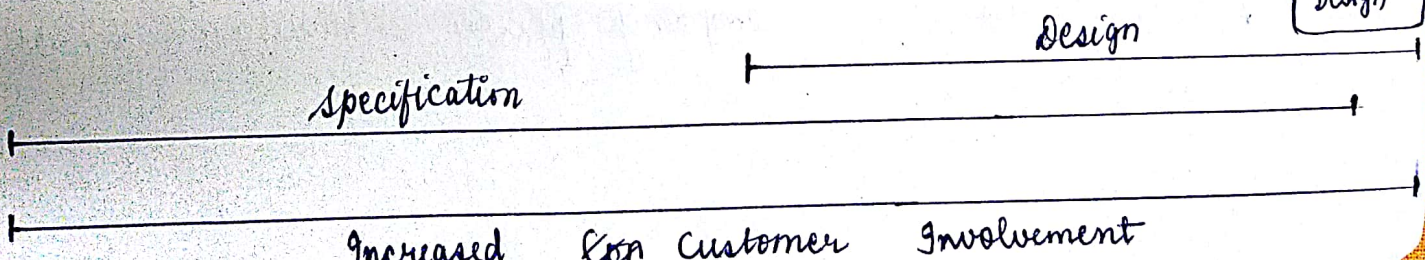
1) Algaabric Techniques : These are used for interface specification.
2) Model Based Techniques : These are used for behavourial specifications.

Formal Methods

These are the collection of techniques used to define

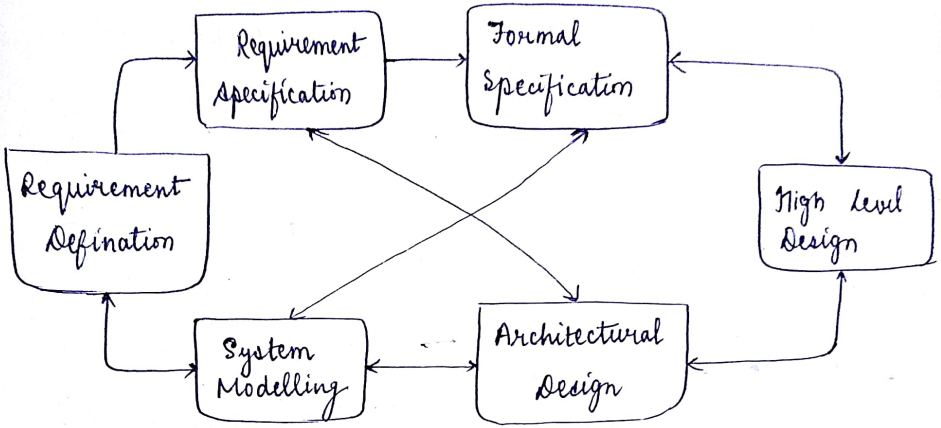
1) Formal specification
2) Specification Analysis and proof
3) Transformation Development
4) Program Verification





specification & Design are mixed. Architectural design is essential to structure the specification. Formal specifications are expressed in mathematical notation with defined vocab, syntax and symmentix.

Specification in Software Process



Explaination : Formal specification traverse between req. specification & High-Level Design. If any change in the formal specification appears it leads to the change in system modelling which is reflected in architectural design. With this repeatative process actual required design is achieved which could further be develop as a system.

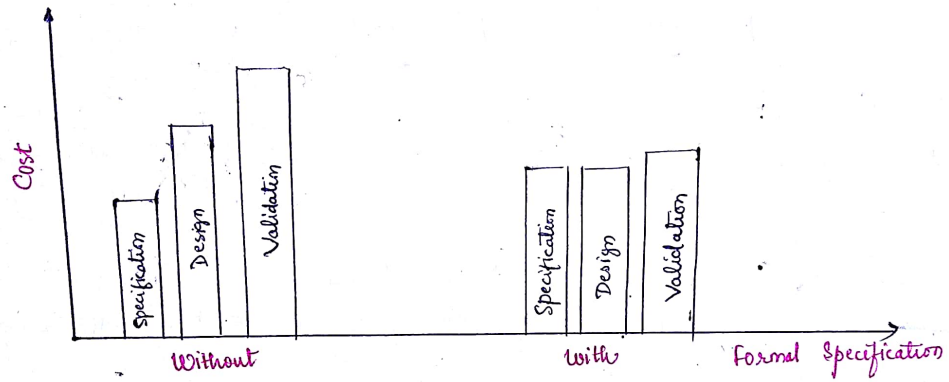
Use Of FORMAL SPECIFICATION

This reduces required error & forces detailed analysis

Incompleteness & Inconsistancy can be discovered & resolved

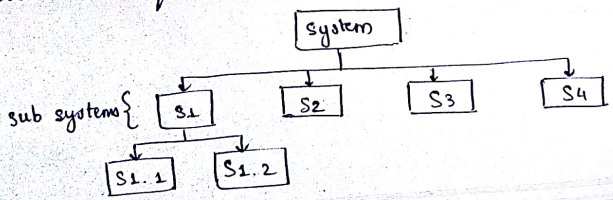
Hence amount of rework is reduced.

⟹ Difference is cost with or without formal specifications



WAYS OF IMPLEMENTING FORMAL SPECIFICATION

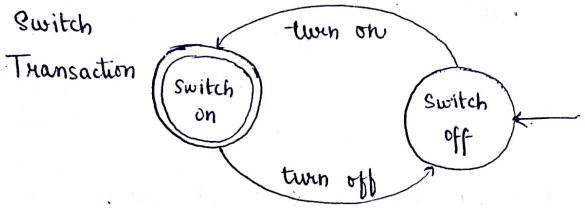
1.↳ **Algaebric Approach**: In this we divide the system into subsystems which consist of minimum dependency on each other. After spliting the system operations on each system is defined & managed accordingly.



2.L> **Model Based Approach**: In this a system is specified in terms of state model that is constructed using mathematical constructs. Operations are defined by modifications to systems state.

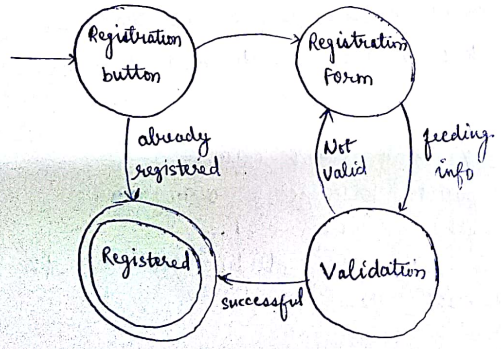
Example of state Transaction Diagram

State Chart diagram are represented by circles
State Transaction by directional arrows.



- No. of State
- No of inputs
- Starting state
- ending state
- set of transaction

: Draw a state Transaction diagram for online registration system



: Create an SRS for online banking system

① 1) Purpose : ① Location independancy ② 24X7 availability

2) Scope : ① Global

3) Definations : ① NEFT :
② Debit :
③ Credit :
④ Transaction Successful :
⑤ Transaction Denied :

4) References : ①* Bank policies *② IT laws

5) Overview : (Upcoming) Handling which kind of users & all, Upcoming functions

② Overall Discriptive :

1) Project Perspective : Dependent on network, payment gateway of third-party ∴ not self contained.

2) Project-Functions : Money transfer, purchase, sell, record logs, Uses bank services (FD, loans), check balance

3) User-Characteristics : Browsing knowledge, Knows his own rights for using this facility, Knows how much is to be paid to bank for using this facility

4) Constraints : Compatibility with other banks
Banking comes under fault tolerance = 0.99999999....
P(fault) = 0.00000001...

5) Assumptions & Dependencies : ① User study SLA ② ~~In~~ less network transaction is not done

③ ① External & Internal Specifications : (Css, php, Java Script) lang
App^n Req, Adaptibility Req.

2) Performance Req : ① OTP in 5 sec
② OTP valid for 15 min
③ Even if 10000 peoples apply it should reply.
④ Validity for min at network gateway.

Constraints
3) Design : ① Font readable ② High vocab english not to be used

VERIFICATION & VALIDATION

Requirement Validation

→ It checks that the right product is built.
→ It makes sure that the product will satisfy its stakeholders.
→ It checks that the requirements meets the goal.

Requirement Verification

→ It checks that the product is built right.
→ Checks that each step is followed properly.
→ It checks consistency & after effects of the system.