# Software Engineering

3/March/2021

2021-02-05

Software → A program which is designed to solve a particular problem to make a task more fast and efficent.

Software → Collection of program

↓ Microprocessor requires software

↓ Algorithm

↓ Instruction

{ Software Engineering → To innovate and create software. }

* Software makes devices to perform task

* Skill, presence of mind is engineering

Programming skills + Logic + Sense → Software Engineering

| * Computer System App | Application Soft |
|---|---|
| → Software which is used to Run The Computer<br>Ex, operating System .. etc | software which are running on computer software are Application soft<br>Ex → VLC ....<br>etc |

* why do we need Software ?

Ans → To use The hardware

* Stack Halder → All The Assets used in developing The software is Stack halder.

→ The Person who keeps the Stack/ or manages The Stack is Stack halder.

# Software development Lifecycle.

1) Requirement → Requirement of Software

↳ Example → Cost of Development, Java Language or any Language.
+ Stack holder Requirement
+ functionality of Software

Requirement

* functional Requirement →

↳ what is functional Requirement? of The Software

⇒ All the Requirements for the Software are functional Requirement

* Non-functional Requirement.

↳ Efficency, Performance, Things like This are non-functional Requirement

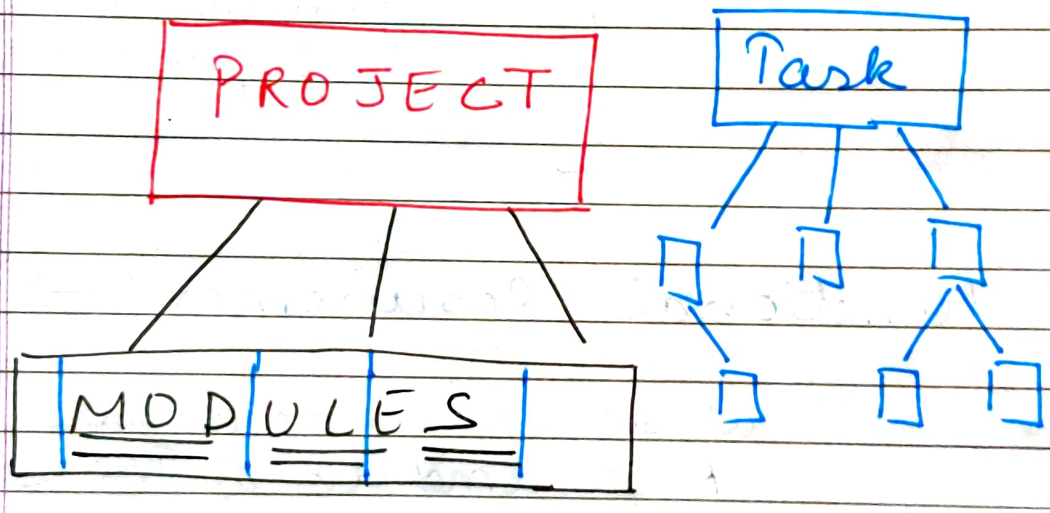
→ Realibility, Performance, Security ... etc.

VIJAYANT
PAGE NO.:
DATE:

# * Nature of Software

1) Properties
↳ 4 nature

→ Complexity →




What is Sub routine ?

→ Software should ~~fulfilll~~ fulfill the functional requirement

* All the Stuff behind the Working of code is called Sub-routine

(*) Sub Routine → Sub Task
me divide karna

→ Modules will lead to complex nature of the Software

2 Nature → conformity

Conformitting → No fixed Rules or principles for development of Software.

* No Rules from physics and other Branches will help in Software development

3) Changibility → Changing of Software without changing functionality of The Software

4) Invisibility → Software is Abstract
→ you can't touch or feel it

Client → Organisation → Requirement
↓
Final Software ← Team Req ← Team meet

2021-02-08

* A software process is the set of activities and associated result that are involved in developing software Products

* It is framework of Software engineering.

* Software Process

→ Used to Built high quality software

* Foundation of Software Engineering

* Every activity which ~~is~~ is used to develop the Project / Software is called Software Process.

—— x —— x —— x ——

# Methods

→ Methods are way {organized ways} which are cost effective and will produce high quality software

→ Method = Tareeka = approach.

a) Communication — Client
b) Requirement analysis — Engineer
c) Design — UI/UX
d) Program construction — Programmer
e) Testing — Testers
f) Support.

* Managers will look into all the requirements

---

Tools → Softwares used

---

A Process Framework →
Overview of all Process.
↓
It is called umbrella activities.
{Final Product}

Generic framework

10/Mar/2021

## Process frame work {Activities}

1) Communication

⇒ create models after

→ Client ke idea ko understand karna

→ Objectives of Software decide karna

2) Planning →

→ Task, Cost, Modules, Resources Sab decide karna

3) Modelling → Create a model {prototype}, Designers

Models include

a) Workflow

b) flowchart

⋮

4) Construction
↳ Start development / Coding

5) Development →
↳ Final Testing
↳ Remove Bugs
↳ Give final Software to client.
↳ Give Support to Client

---

* All the above core activity, there are Sub activity

* Communication will go on ~~at~~ till end of the Project.

⇒ Sub Process → All activity have Sub process.

→ Ex → Communication
- Team
  - Leader
  - Client
- Client

Members

# Umbrella Activities

* All the activity in the Process which give final Result are umbrella Activities

| Minor Act | + | Major Act | = | Umbrella Act |
|---|---|---|---|---|

1) Software Project tracking and control.
   {Manager's Task} ↳ Planning and timeline + quality + Suggestion

2) Risk: → Risk that can affect the outcome of project should be managed.

3) Software quality assurance.
   ↳ Keep a check on all activity to ensure quality

4) Tech Reviews → To ensure Quality work and remove error

5) ~~Acc~~ Measurement → Measure to make sure that Stakeholder demand are met.

6) Software Config: Management of changes {new}

7) Reusability management: Project can be reused.

8) Work product preparation and Production:
↳ Models, workflow flow chart .... etc.

---

→ Adaption of Process Model

⇒ Agile and adaptable ||.
↳ Projects can be done in fast mode.

* There are several Process model.
* One has to select effective model from them.

* planning ke Baad decide karte process model.

VIJAYANT
PAGE NO.:
DATE:

# * Agile Models

↳ Many Modules are developed together.

12- Feb/ 2021

PPT se Kiya [No Audio]

{CMMI MODEL

* Industry framework

Capacity
Maturity
Model
Integration

↳ Improves Quality
↳ Hardware Software mix
↳ Improves Business Result.

* Initial → Managed → Defined
↓
Optimizing ← Quantitative

VIJAYANT
PAGE NO.
DATE:

|| 2021-02-22 ||

* CMMI
  ↳ Bring The Project to ISO model

* 5 Models { 5 Labels }

⇒ ~~Labels~~ Label-I { Initial }

* poorly controlled
* unpredictable
* reactive

} noobs

⊕ Lack of Exprience.
⊕ Performance may not be up to The mark.
⊕ poor quality, cost, schedule.
⊕ Software will be functional.

⇒ Level-2 { Managed }

* Planned
* Skilled people
* Adequate Resource
* Regularly Reviewed
* Well planned, performed, monitor
* Quality is better than level 1.
* Controlled Outputs.
* Well Documented
* Well scheduled.

{ Better Than level-I }
[Managed]

[All The Tasks are Better than level-I. Schedule, Cost.... etc]

⟹ Maturity level - 3 {Defined}

* Startups will be called organisations at level-3

* Process should be well defined

* Well understood, well deployed and well executed

* Everything should be managed

* One can't opt-out from a project if he starts working in that.

⟹ Maturity level - 4 {Defined} {Quantitatively managed}

* ISO certificate
* Reputation of Brand
* More controll over project and processes.

⟹ Maturity level - 5

* ~~Also~~ Always Improvement is the aim.

CMM Level = Accountability + Responsibility ↑

## CMMI Components

→ Process Area ⇒ This is a must process that need to be implied acc to the maturity level.

⇒ Maturity 2 → 7 Process Area

⇒ " 3 → 11 Process Area

* Each process area has goals which need to be achieved.
* Goals have some practises.

---

## Process Description {include}

* Product
* Roles {People involved}
* Pre and post-condition {what has changed}

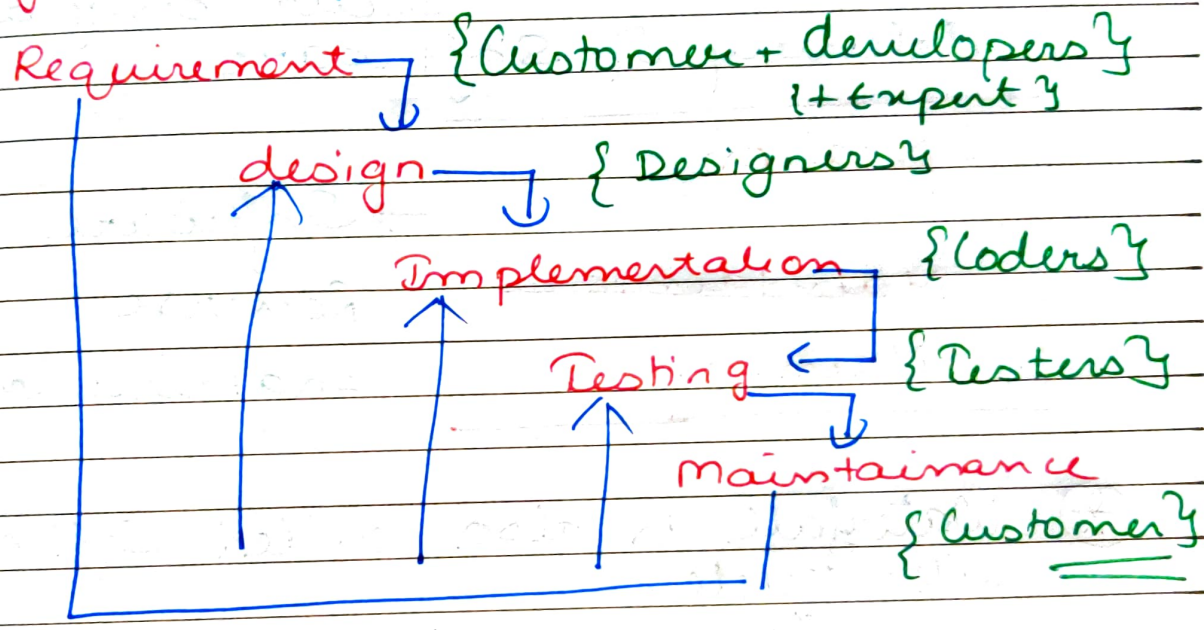
| Plan driven | Vs | Agile Process |
|---|---|---|
| * Software is curated with a plan | | * Software is curated with Somewhat plan |
| * First planning Then development is started | | * No full planning |
| | | * According to customer requirement Software is changed |

- We can use ~~both~~ both agile and plan-driven together.
- There are no right or wrong process.

## ⇒ The Waterfall Process {The Plan driven}

* Top to down {Jahan Bohat time ho}

~~Regu~~




* Module wise testing are done
* Integeration ke Baad phir Testing.

⇒ **DrawBacks**

* Difficult to make changes

2021-02-24

Process → Collection of Activities

* Waterfall Process → changes can occur only after delivery of Software.

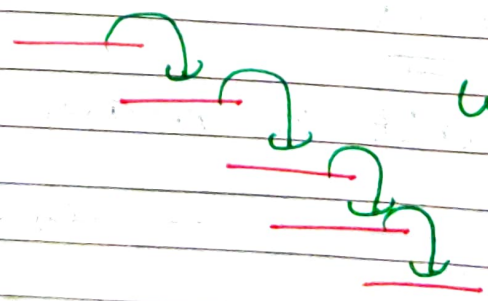
Operation
&
~~Maintainance~~
Maintenance

→ Software Reverse engineering

* Waterfall is also called Software life-cycle.

* Oldest process

* The Product is delivered after linear sequence.



Waterfall

## Limitations

* A phase should not start until the previous phase is signed of
* Changes are costly and involve significant work.
* Software is put into use during final phase of Study
* Partion of Modules is not possible
* Requirements should be well understood.

---

## Evolutionary Model ??

* Assignment {Look into Models}

|| 2021-02-06 ||

→ **Personal Process Model**

* The activities you do while developing a personal project are called personal Process model.

→ **Team Process Model**

high-quality
skills + discipline
with Budget

The activities we have to do to while developing a Team project. It is a team process model

* To ~~deliver~~ deliver a high quality Product

**Personal Software Process**

* Manage the quality of their project
* Make commitments they can meet
* estimate and planning
* Reduce defects in their products.

# Team Process Model

* Improves Team performance
* Team discipline
* High quality Software
* Secure Products
* Process Management in the team

## c) Engineering TeamWork

* College project are usally done in team
* Skilled Member in team
* hardware and Software by individual
* Highly Effective.

* Effective Team: There should be skilled members.
→ goal should be clear.
→ Adequate Resources should be There.

VIJAYANT
PAGE NO.:
DATE:

## What agile?

* fastest nibba alive
* create something of good quality within less time
* to change the Software together.

⇒ Agile Software Development

* We can do multiple changes in the life cycle of the Software.
* changes are iterations
* flexible software.

⇒ Properties of Agile SD.

* Light-weight methodology
* Small to medium sized team
* vague and changing requirement
* changing Techniques
* Simple design
* Minimal System into production

VIJAYANT
PAGE NO.
DATE:

## ⇒ Characterstics of Agile SD.

* Modularity {Divide into Modules}
* iterative {changes}
* Time Bound {Time}
* Incremental {Daily Development}
* convergent {Should work in goal}
* People-Oriented {all cooperation}
* Collaborative {Skills sahke ++}

## ⇒ Existing agile method.

* Extreme Programming (XP) {gyan in coding} → {2 coders in same code}
* Agile Unified Process {All the process in Agile model}
* Scrum {Process that can done in agile}
* changes are made on the Spot

→ Changes karna are scrum

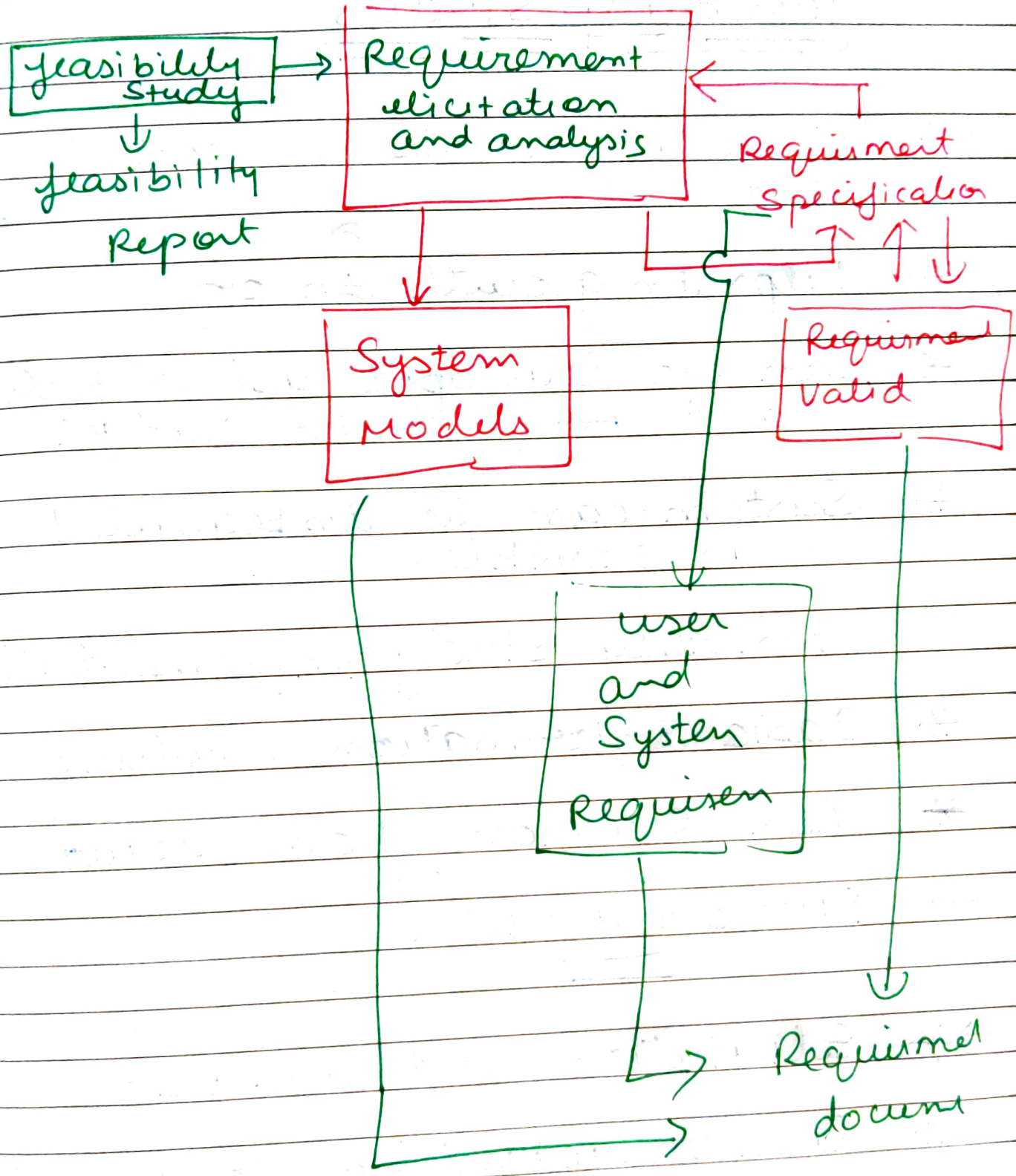
2021 - 03 - 01

# Requirement Engineering

[Self - Validation Steps]

* First Step in Software Development
* Requirement elicitation
  {Client tells Requirement}
* Requirement analysis
  {Analysis of all the Requirement of the client}
* Requirement Validations
  {Validating all the Requirement}
* Requirement Management
  {flow chart + Resource Manage}

VIJAYANT
PAGE NO.
DATE:



# Feasibility Studies

* Check if the SD is feasible or not
* helps in growth ~~in~~ of org
* Can be developed using tech in hand ~~in~~ and in budget
* System can be integerated is possible.

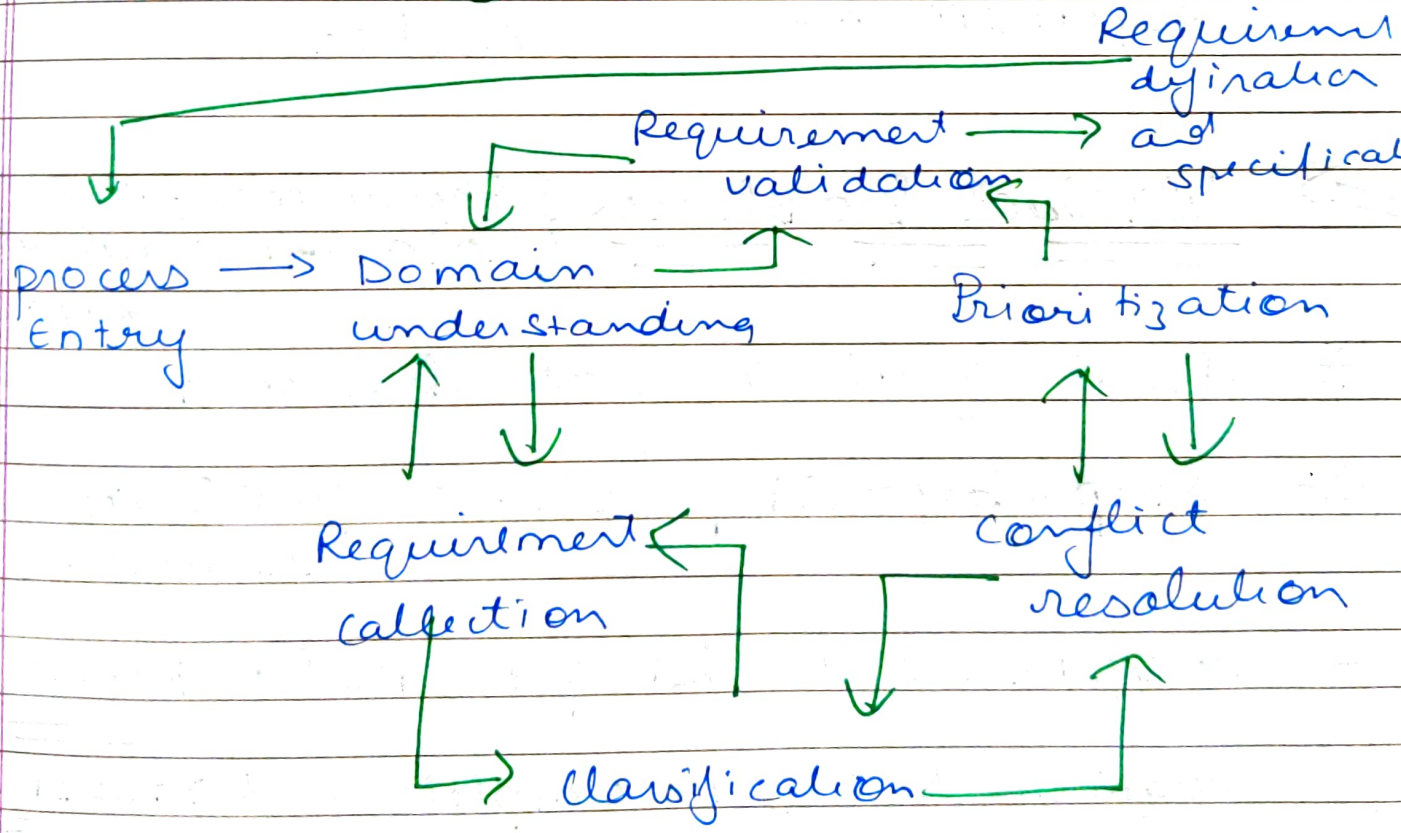
Feasibility Study implementation

* Basic information analysis
* Ask question from yourself
* Question in pic

→ * Elicitation and analysis
* Question regarding domain, Services, tech required
* System's operational constraint
* include → end user, stackholder trade union.

# Problems of requirement analysis

* Stakeholders don't know what they want
* Stakeholder express requirement in their own terms
* Organisational and political factors may influence the system requirement
* Stakeholder express requirement in their own terms.




Requirement analysis process

* → Requirement can be according to stakeholders too.

|| 2021-03-03 ||

1) Requirement defination ⇒ Clients

2) Requirement Specification ⇒ designer
engineer
developer

3) Software Requirement ⇒ client engineers, coder.

⇒ Wicked Problems {Shuru me pta nhi hota ki final goal kya hai

* final goal is not known

* step-by-step | phase goal is released

* requirement ka pta nahi Rehta.

* requirement are incomplete and incosistent.

* Requirement Should be according to the Client need.

VIJAYANT
PAGE NO.:
DATE:

# Reasons for inconsistency

* Irregular
* partial requirement ko fulfill karna at a moment.

* Compromise in requirement due to many users requirement

* Need to create a prototype for requirement clarification.

Documentation {SRS}
↳ Software Requirement Specification

* Fill ~~to~~ up all the Steps used in The Software

* All requirements and activity

Template

1) Intro

2) Glossary {Technical Terms used}

3) System Model

4) Functional requirement definition

5) Non-functional requirement

6) System evolution { growth }

7) Requirement Specification { Details }

8) Appendices { Data Base + hardware }

9) Index

⟹ Validation

↳ To valuate That our requirement is feasible [Validation Karna]

⟹ Verification → That our app is working good with Software

[Sab sahi se chal Raha ki nahi]

(A) Validation { Requirement

1) Validity

2) Consistency

3) Completeness { sab hai ki nahi }

4) Realism { ho SAKTA ? }

# Requirement Evolution

* Requirements Evolves
* So plan for changes.

Initial understanding of problem → Initial Requirements

Changed Requirement of problem → Changed Requirement

→ Time

## Requirement Classes

1) Enduring requirement { Jo imp ho }
2) Valatile Requirement { Jo itni imp nahi ho }

# Controlled Evolution

1) Requirements document $V_1$

↓

System Implementation → S.I

$V_1$ $V_2$

{due to change in requirement}

after implementation

2) Requirement document $V_1$ → Requirement Document $V_2$

↓ ↓

S.Imp $V_1$ → S.Imp $V_2$

Initial analysis Model → high level view
↓
design objects
↓
Implementation

* Requirement themselves change
* our view | model of them change.

* Always try to find what your client wants from your software

* Requirement elicitations
{ client's interogation

+ Finding domain + constraint + maintenance
+ Stakeholder requirement.

* System Model

1) Different Models can be used in analysis activity

2) Partioning → Identifies the Structural

Abstraction

Projection

* different Bande different Soch.
* full analysis karo.

* Try to take views of many people as possible
  * different types of Stakeholders
  * different views among Stakeholder.

* Multi - perspective analysis
  ↳ No Single way to analyze the System

⊕ Ek problem, many views
↓
Sabka analysis karo

View Point

1) Data Sources {kon data use karega}
2) Representation frame work {kaise framework pe banegi}
3) Receiver of Services
{