Bidding ⇒ Process in which organizations or customers tell what they want.

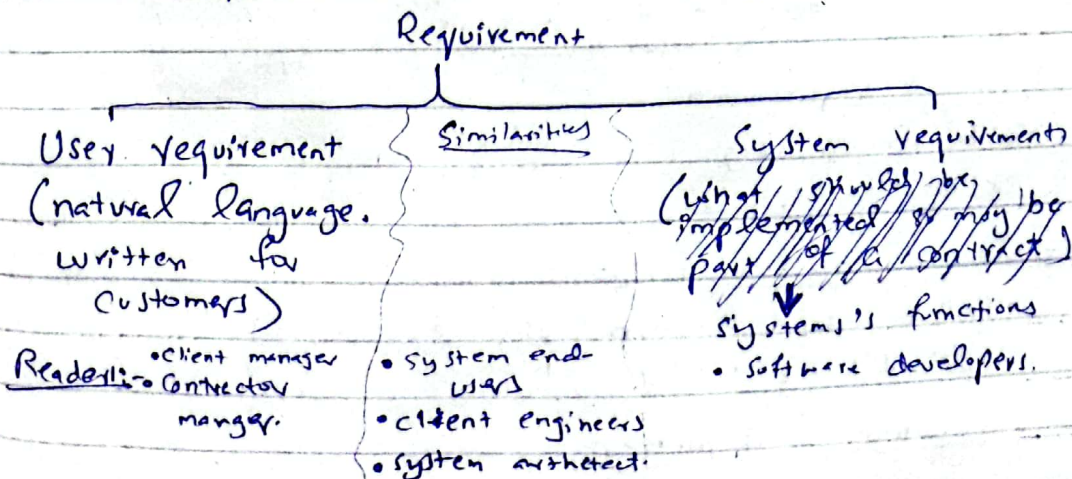# Chapter no 4. Requirements Engineering

⇒ **Requirements Engineering :-**

The process of establishing services that the customer wants from a system and had restrictions under which it operateves and is developed.

⇒ **What is requirement?**

• User can give a statement or a mathematical expression

○ **Requirement** serve as dual function

Requirements {
  • — Basis for bid for a contract ⇒ must be open to interpretation. User developer leo bati skey
  • — Basis for Contract itself ⇒ defined in detail
}

⇒ **Abstract requirements:-**

• Company should create abstract requirements. Contractor kush avr beh daal dae gr. avr phy document ban jaege.

Requirement
├─ User requirement (natural language. written for Customers)
│   Readers:- • Client manager • Contractor manger.
├─ Similarities
│   • System end-users
│   • client engineers
│   • system arshitect.
└─ System requirements (what should be implemented & may be part of a contract)
    system's functions
    • Software developers.

⇒ System Stakeholder :-
Any person or organization who is affected
by the system in some way

| End users | System manager | System owner | External Stakeholder. |
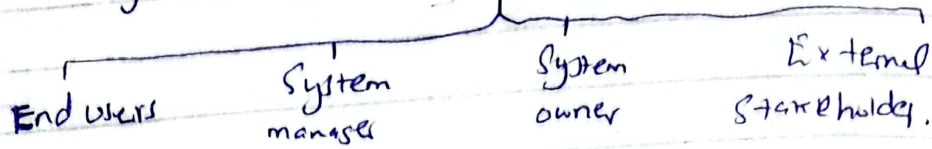
⇒ Agile method and requirements :
- do not make them detailed b/c they are agile.
- requirement document is out of date.
- Agile method use incremental requirement engineering.

⇒ Functional requirement   Non-Functional requirement
- input jo be h   - quality - standards
  aus pr kaam   - If not quality good,
- Auv kia nahi krna   system useless
  wo beh likha   - security → quality and
  hoga.   it should
              Domain requirements   be implement
                                               by function.
      - Any type of software domain
        but it should well work
        for that domain.

Functional requirements.
- which functionality
- depends on user.

⇒ Requirements imprecion:
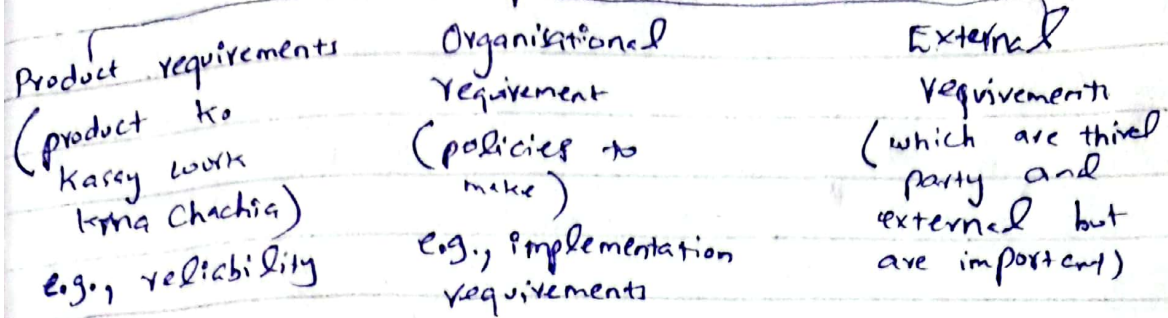- Problem arise when user do not tell
  developer precisely.

Requirements Completeness:-
- include description of all facilities
Requirements Consistency:-
- no conflits in description.

Non-Functional Classification

=>

| Product requirements (product to Kasey work krna Chachia) e.g., reliability | Organisational requirement (policies to make) e.g., implementation requirements | External requirements (which are third party and external but are important) |

=> <u>Useability requirements:</u>
- easy to use and errors less.

=> <u>Metrics</u> for specifying non-functional requirements

<u>Properties:</u> speed, size, Ease of use, Reliability, Robustness, Portability.

↓

Time to restart after failure.

\# <u>Requirement's engineering processes</u>

- ✓ depends on application + domain, people involved and Organisation

- Common Processes for generic are:-
  - - Requirements elicitation → process of gathering.
  - - " analysis
 - - " validation
  - ○ - " management

① <u>Requirement's elicitation</u> / <u>Requirements Discovery</u>
- customers & technical staff find domain of application. May involve stakeholders.

<u>Stages:-</u>
① Requirements discovery (Range of stakeholders)
② " classification & organization
③ " prioritization & negotiation
④ " specification

Problems:-

• Stakeholder don't know what they want.

```
                    → discovery →
Specification                        Classification
         ↖  prioritization  ↙
```
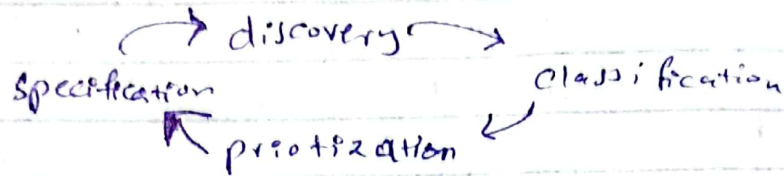
⇒ **Interviewing**

Closed interviews
based on
(pre-determined
question)

Open interviews
(issues are
explored
with stakeholders)

⇒ **Ethnography**

• (Research method used in social
and cultural anthropology, sociology)

• Work is richer or more complex
shown by enthrographic studies
but some simple models do not
tell right.

Scope:-

• Effective for understanding
existing process but can not
identify new features.

Focused Ethnography

• Combines Ethnography to prototype.

Scenarios:- (Structured form of user story)

- Description of starting situation
- Description of normal flow of events.
- Description of what can go wrong.
- " " other activities
- " " scenario finishes.

Requirements specification:

- more accurate and detailed • It should be as complete as possible.
- Ways of writing (NL, structural.NL, design description language, graphical notations, mathematical specifications)

⇒ •— Structured specifications → • Requirements are written in standard way.

⇒ •— Form-based specifications
  ① Definition of function
  ② " where inputs came from
  ③ " where output will go
  ④ Information needed for computation
  ⑤ Description of action
  ⑥ Pre post condition.
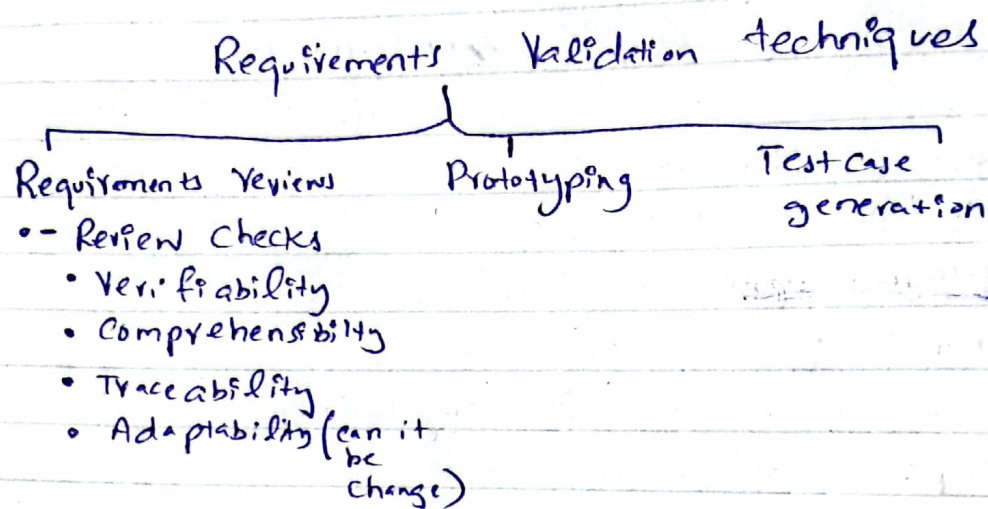 ⑦ side effect of function.

⇒ •— Tabular specification
  • used for supplement NL.

Condition	action

⟹ **Use cases:-**
   ATM vali example.

⑥ **Requirements Validation:-**
   (·errors ko nikal kr, aur user
   sa pooch lo kay yahi hain
   requirements ya nai.)

⟹ **Requirements Checking:-**
   Realism → Can requirements be implemented
          on avalible budget.

   Requirements   Validation techniques

   Requirements Reviews   Prototyping   Test Case
   ·- Review Checks   generation
     • Veri fiability
     • Comprehensibilty
     • Traceability
     ∘ Adaptability (can it
                     be
                     change)

⟹ **Requirements Change:-**
   • Technical environment of system always
     change after installation
   • People who pay and users are rarely same
     people.
   • Large systems had diverse user community.
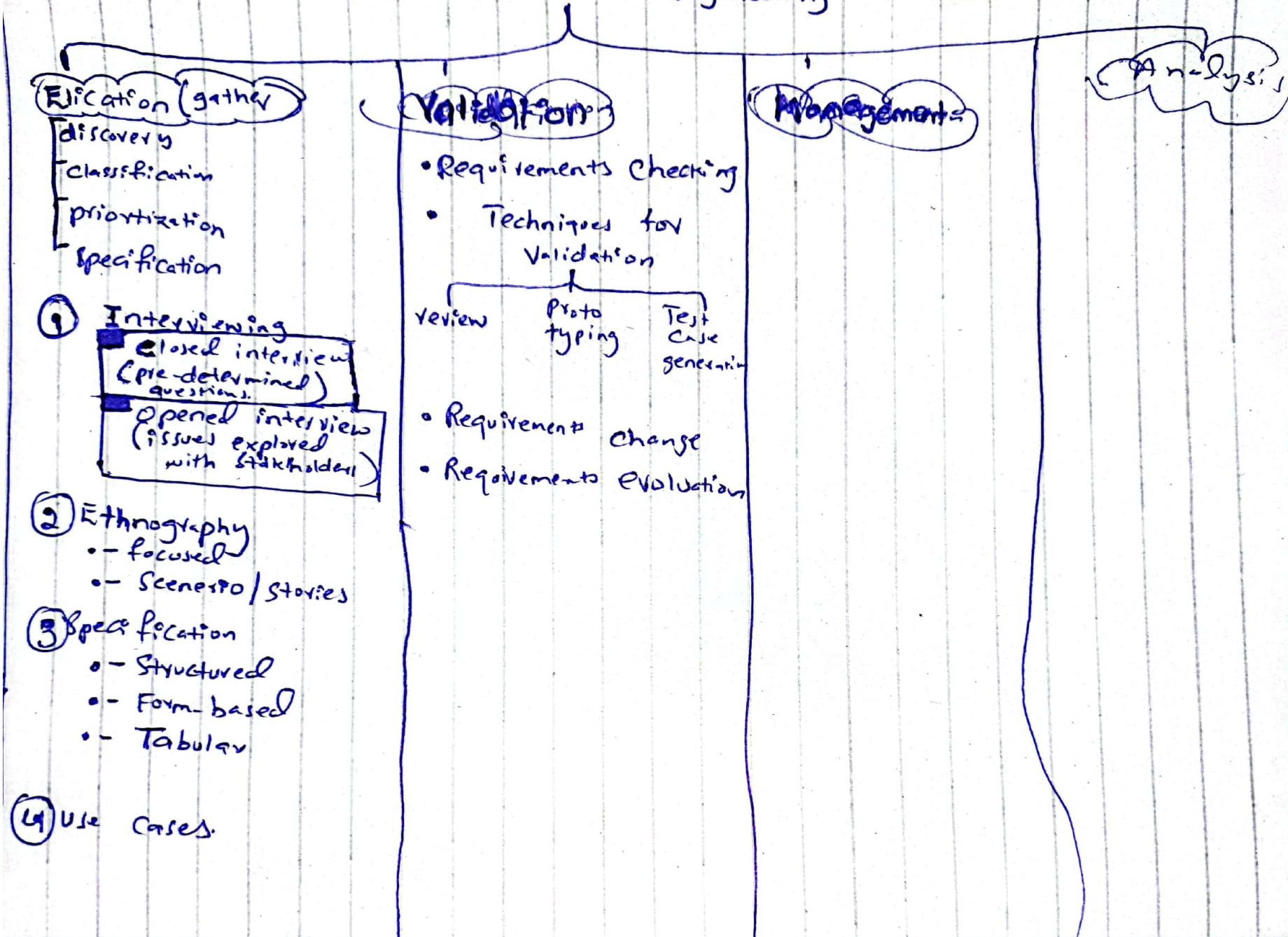
⟹ **Requirements evoluation:**

③ Requirements management:-

(Manage to change requirements during process)

Planning:-
- Requirements identification
- A Change management process
- Traceability policies
- Tool support.

Requirement Engineering Process.

Elicitation (gather)
- discovery
- Classification
- priortization
- Specification

① Interviewing
- closed interview (pre-determined) questions.
- Opened interview (issues explored with stakholders)

② Ethnography
- – focused
- – Scenerio/Stories

③ Specification
- – Structured
- – Form-based
- – Tabular

④ Use Cases.

Validation
- Requirements Checking
- Techniques for Validation

   review | Proto typing | Test Case generation

- Requirement Change
- Requirements evolution

Management

Analysis

- System modeling is process of developing abstract model of system.
  (UML)
- We use models which are used in requirements engineering

➡️ System Perspective:
  - External (environment of system)
  - Interaction (model interactions b/w system and environment)
  - Structure (organized)
  - Behavioral (responds or behaviour to events)

➡️ UML Diagram Types:-
  - Activity diagram (activities involved in process)
  - Use case (interaction b/w system & environment)
  - Sequence (actors, system)
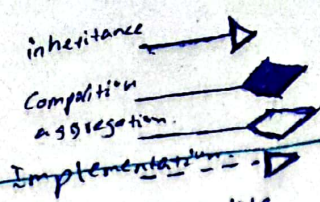 - Class Diagram
  - State Diagram

① Context Model :-
  - What lies outside system boundaries
  - System boundaries (what are inside and outside system)

  | <<System>> |

  | <<System>> | | <<System>> |
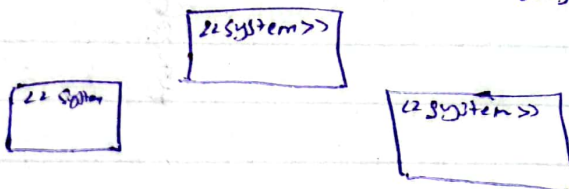
② Interaction Model: (different component or object communicate and exchange information)

a) Use case diagram

inheritance →
Composition ————◆
aggregation. ————◇
Implementation --→

- Transfer data ule case:- (involves movement of data from one location to another, ensuring data Security)

b) <u>Sequence diagram,:</u>

③ <u>Structural Model:-</u>
- class diagrams.
- generalization. (inheritance)

④ <u>Behavioural Model:-</u>
(System kay against koi cheez aye hy, tou aus pr system kia act lega ga.)

[• - Data (Some data arrives)
  • - Events (Some event like alarm and what are System is doing)

Data-driven modeling (design on basis of data)
<u>Event-driven modeling</u> (how system responds to external and internal events)
<u>State machine models.</u> (when event occur State Changes)

⑤ <u>Model-driven Engineering(MDE)</u>
(Shifts focus from coding to high-level model which are than transformed to excutable program

┌ • <u>Model-driven architecture (MDA)</u> (creating models Creates models and Convert it to coding.
top
level a) Computation independent model (CIM) / Domain model
  non-technical, • important domain abstractions.
           • tell about high level business goals.
inter
medim b) Platform independent model (PIM)
           intermediate.
low c) Platform specific model (PSM) (close to implementation, used to generate code.

Tashfeen Abbasi / Laiba Mazhar
NUCES, FAST, ISL
abbasitashfeen7@gmail.com