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

Chapter no 4 Requirements Engineering

⇒ Requirements Engineering :-

The process of establishing services that 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
  - Basis for bid for a contract ⇒ must be open to interputation. user developer ko bata skay
  - Basis for contract itself ⇒ defined in detail

Requirements

⇒ Abstract requirements:

- Company should create abstract requirements. Contractor kush aur beh daal dae gr. aur phr document ban jaega.

Requirement

User requirement (natural language. written for customers)	Similarities	System requirements (what should be implemented & may be part of a contract)
Readers:- • client manager • Contractor manger.	• System end-users • client engineers • system architect.	↓ Systems'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- requirement document is out of date.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

- input jo bolo
  aur pr kaam
- Aur kia nahi krna
  wo bolo likha
  huga.

Non-Functional requirements

- quality - standards
- If not quality good, system useless.
- security → quality and it should be implement by function.

Domain requirements

- Any type of software domain but it should well work for that domain.

Functional requirements.

- which functionality
- depends on user.

⇒ Requirements imprecision :-

- Problem arise when user do not tell developer precisely.

Requirements Completeness :-

- include description of all facilities

Requirements Consistency :-

- no conflicts in description.

Non-Functional Classification

=>

- **Product requirements** (product ko kasey work karna chachia) e.g., reliability
- **Organisational Requirement** (policies to make) e.g., implementation requirements
- **External Requirements** (which are third party and external but are important)

=> **Useability requirements:**

- easy to use and errors less.

=> **Metrics for specifying non-functional requirements**

Properties: speed, size, Ease of use, Reliability, Robustness, Portability.

↓

Time to restart after failure.

Requirement's engineering processes

- depends on application domain, people involved and organisation
- Common Processes for generic are:-
  - Requirements elicitation → process of gathering.
  - " analysis
  - " validation
  - " management

Requirements elicitation / Requirements Discovery

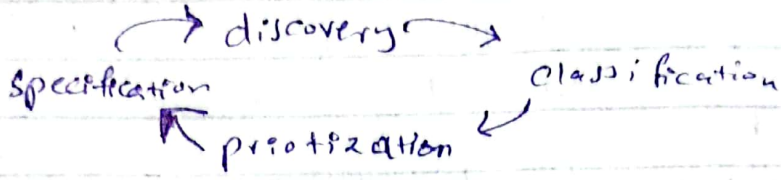
- customers & technical staff find domain of application. May involve stakeholders.

Stages:

1. Requirements discovery (Range of stakeholder)
2. " classification & organization
3. " prioritization & negotiation
4. " specification

Problems:-

- Stakeholder don't know what they want.



⇒ 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 enthnographic studies but some simple models do not feel 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 wali example.

6 Requirements Validation:-

(Errors to nikal kr, aur user sa pooch lo kay yahi hain requirements ya nai.)

=> Requirements Checking:-

Realism → Can requirements be implemented on available budget.

Requirements Validation techniques

- Requirements Reviews
  - Review Checks
    - Verifiability
    - Comprehensibility
    - Traceability
    - Adaptability (can it be change)
- Prototyping
- Test case generation

=> 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 evolution:

(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
  - prioritization
  - specification

1. Interviewing
   - closed interview (pre-determined questions)
   - opened interview (issues explored with stakeholders)
2. Ethnography
   - focused
   - Scenario / stories
3. Specification
   - Structured
   - Form-based
   - Tabular
4. Use Cases.

- Validation
  - Requirements Checking
  - Techniques for Validation
    - review
    - Proto typing
    - Test Case generation
  - Requirements Change
  - Requirements evolution

- Management
- Analysis

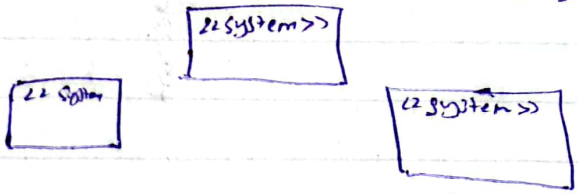
Ch no5 System Modeling

- 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 of interactions blw system and environment)
- Structure (organized)
- Behavioral (responds or behaviour to events)

⇒ UML Diagram Types:-
- Activity diagram (activities involved in process)
- Use case (interaction blw 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)



② Interaction Model: (different component or object communicate and exchange information)

a) • Use case diagram.

inheritance
Composition
aggregation
Implementation

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

b) Sequence diagram:

③ Structural Model:-
- class diagrams.
- generalization. (inheritance)

④ Behavioural Model:-
(system kay against koi cheez aye hy, tou aus pr system kia act leana ga.)
- Data (some data arrives)
- Events (some event like alarm and what are system is doing)

Data-driven modeling (design on basis of data)

Event-driven modeling (how system responds to external and internal events)

State machine models. (when event occur state changes)

⑤ Model-driven Engineering (MDE)
(Shifts focus from coding to high-level model which are than transformed to executable program)

- Model-driven architecture (MDA) (creating models)
  Creates models and convert it to coding.

a) Computation independent model (CIM) / Domain model
Non-technical, important domain abstractions. tell about high level business goals.

b) Platform independent model (PIM)
intermediate.

c) Platform specific model (PSM) (close to implementation, used to generate code.)