# UNIT-II

## REQUIREMENT ELICITATION, ANALYSIS AND SPECIFICATION

① Requirement elicitation is the practice of collecting the requirements of a system from users, customers and other stakeholders. The practice is also sometimes referred to as "requirement gathering".

② Types of Requirements -

(i) Functional Requirements -

→ describe all the required functionality or system services.

→ dependent upon the type of software, expected users & type of system.

Problems -

→ Requirement imprecision

→ Requirement completeness and consistency.

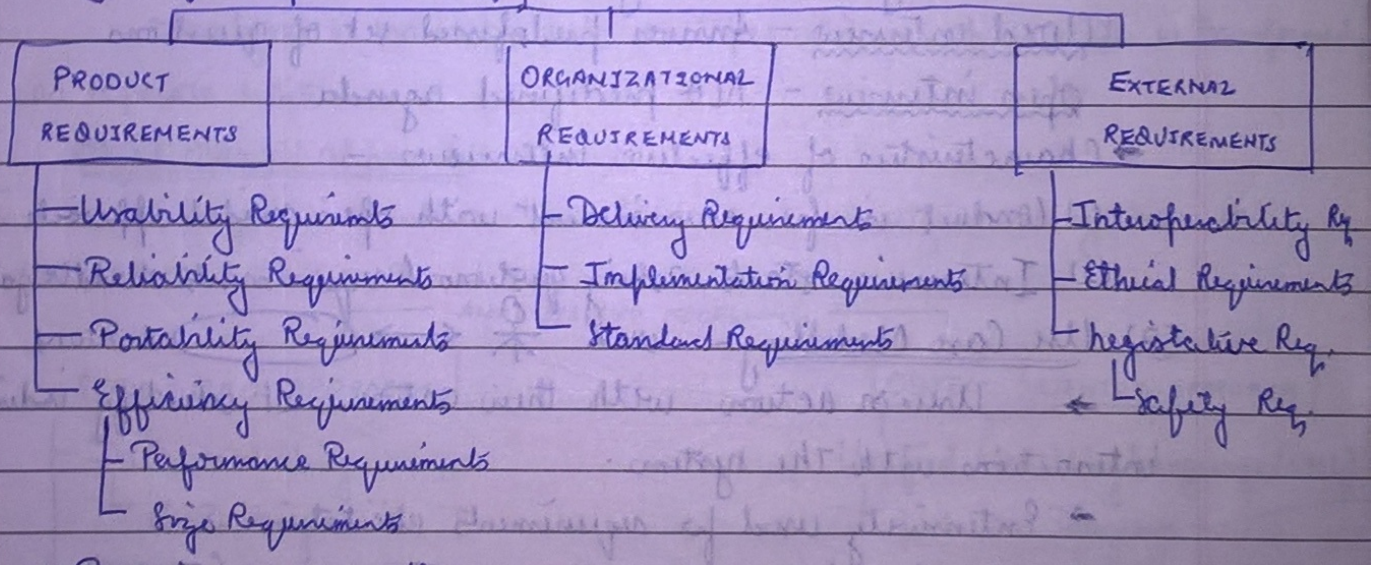
(ii) Non-Functional Requirements -

→ describe system properties and constraints.

→ System properties such as reliability, response time, storage requirements

→ Constraints such as input/output device capability, system representations etc.

→ Types of non-functional requirements -




→ Requirement arise through -

(1) User needs

(2) Because of budget constraints

(3) Organization policies

(4) Need for interoperability with other software or hardware systems.

(5) Because of external factors such as safety regulations.

(ii) User Requirements -

It should describe functional and non functional requirements in such a way that they are understandable by system users who don't have detailed technical knowledge. Eg:- natural languages, tables, diagrams

③ Requirement Sources and Elicitation Techniques -

~~Requirement Sources~~ - Most commonly used technique is to conduct meetings

(1) Stakeholders - like end user, system maintainence engineers etc

Problems -

→ Unrealistic expectations

→ Differences in the requirements (different stakeholders)

→ Economic and business environment

→ Political changes.

(2) Interviewing -

Communicate the stakeholders by asking them various questions about the system. Two types -

Closed interview - Answer predefined set of questions

Open interview - No predefined agenda.

→ Characteristics of effective interviews -

(1) Conduct in free environment with open minded approach.

(2) Interviewer starts asking questions & requirements should be gathered by

(3) Use Case Modelling - USER ⟷ ACTION

→ User or actors with their actions. It identify individual interactions with the system.

→ Extensively used for requirements elicitation.

(4) Facility Application Specification Techniques (FAST)

It is an approach in which joint team of customers and developers work together to identify the problem, propose elements of solution, negotiate different approaches and prepare a specification for preliminary set of solution requirements.

Guidline for FAST approach → make list of objects, services and constraints

(5) Quality Function Development (QFD) -

It is a quality management technique which translates the customer needs and wants into technical requirements.

There types of requirements in QFD are -

(1) Normal Requirements → as per goal & objectives of the system.

(2) Expected Requirements → software package.

(3) Exciting Requirements → Satisfied by software beyond customer's expectation

(4) Analysis and Modelling for function and object oriented design -

Analysis modelling is a technical representation of the system.

→ Analysis model objectives -

→ To describe what the customer requires.

→ To establish a basis for the creation of a software design.

→ To derive a set of valid requirements after which the software can be built

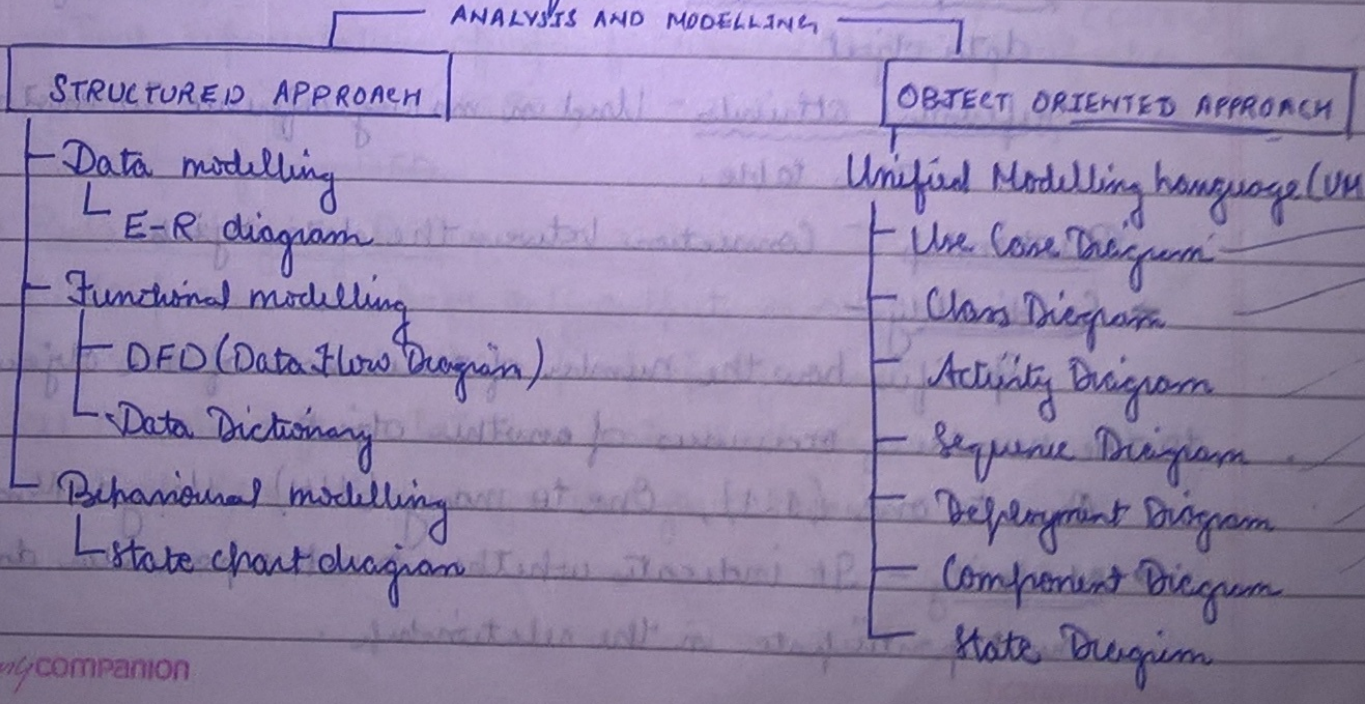
→ Analysis modelling approaches -

Structured approach -

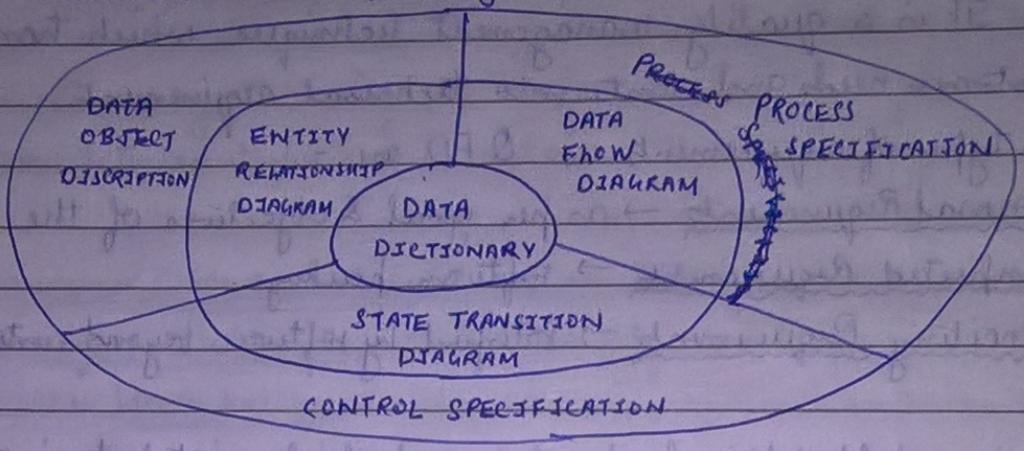
Analysis is made on data and process in which data is treated as separate entities.

Object oriented approach -

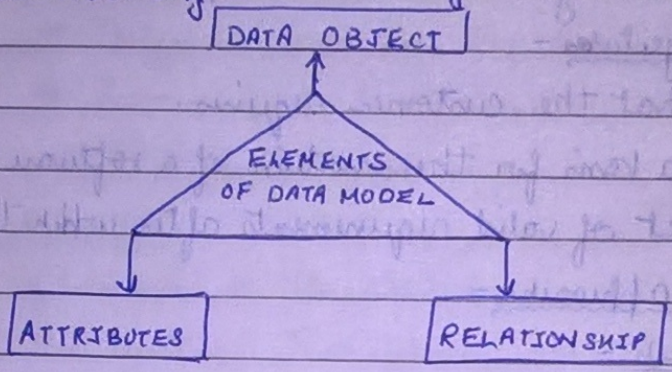
Analysis is made on the classes and interaction among them in order to meet the customer requirements.



→ Basic structure of analysis model -




→ Data Modelling - data objects are examined independently of proces




Data Object - set of attributes (data items) that act as an aspect, characteristic, quality, or descriptor of the object

Attributes - define the properties of data object. Three types -

Naming attributes - Used to name an instance of data object

Descriptive attributes - Used to describe characteristics or features of data object

Referential attributes - Used in making reference to another instan in another table.

Relationship - Connection between the data objects

→ Cardinality -

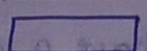
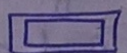
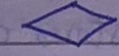
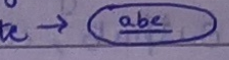
Specifies how the number of occurences of one object is relat to the number of occurences of another object

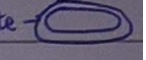
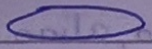
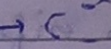
One-to one (1:1), One to many (1:N), Many to Many (M:N)

→ Modality - It indicates whether or not a particular data object must participate in the relationship.

→ Entity-Relationship Diagram - (ERD)

Graphical representation of object relationship pair. Components of ERD are -

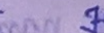
Entity → 
Weak entity → 
Relationship → 
Key attribute → 

Multivalued attribute → 
Attribute → 
Derived attribute → 

→ Functional modelling -

Used to represent the flow of information in any computer based system. Three generic functionalities are input, process and output

→ Data Flow diagram (DFD) -

Process →  Data store →  Flow of data = →

External Entity → 

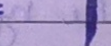
→ Represents the system at any level of abstraction.

→ DFD can be partitioned into levels that represent increase in information flow and detailed functionality

→ level 0 DFD ⇒ 'fundamental system model' or 'context model'

level 1 DFD = sub functions of overall system.

→ Control Flow Diagram - show control flows

Control flow =  Window =  (CSPEC)

→ Two representations of specifications are Control specification and process specification (PSPEC)

~~Rules of defining DFD~~

→ Rules of dinging DFD -

(1) No process can have only outputs or only inputs

(2) Verb phrase should be ~~for~~ properly used

(3) No direct flow between data flows and external entity.

(4) Data flow should go through a process.

→ Data Dictionary -

→ It can be defined as an organized collection of all the data elements of the system with precise and rigorous definitions so that user and system analyst will have a common understanding of inputs, outputs, components of stores and intermediate calculations.

→ Stores information like Names, alias, where-used or how is used.

Notations - Composition (=), Sequence (+) AND, Selection (|) OR, Repetition ($\{\}^n$), Optional data → (), *...* comments.

→ Automated (CASE) tools can be used to maintain the data dictionary.

Advantages -

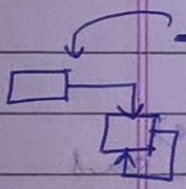
(1) Support name management and avoid duplication.

(2) Store of organizational knowledge linking analysis, design and implementation.

→ Behavioural Models -

Describes overall behaviour of a system represented by state chart diagram.

→ State chart diagram - Collection of states and events. The event cause the system to change its state.



It also represents what actions are to be taken on occurrence of particular event.

(5) System and software requirement specification -

→ Software Requirements Specification (SRS)

Software requirements provide a basis for creating the Software Requirements Specification (SRS).

SRS is useful in estimation cost, planning, team activities, performing tasks and tracking the team's progress throughout the development activity.

DOCUMENT TITLE

Author(s)

Affiliation

Date

Document Version

1. Introduction
   - 1.1 Purpose of this document
   - 1.2 Scope of this document
   - 1.3 Overview
2. General description
3. Fuctional Requirements (Descriptions, Criticality, Technical Issues, Cost and Schedule), Risks, dependencies with other requirements)
4. Interface Requirements
   - 4.1 User Interfaces
     - 4.1.1. GUI (Graphical User interface)
     - 4.1.2. CLI (Command line interface)
     - 4.1.3. API (Application Programming interface)
   - 4.2. Hardware Interfaces
   - 4.3. Communication Interfaces
   - 4.4. Software Interfaces
5. Performance Requirements
6. Design Constraints
7. Other Non-Functional Attributes
   - 7.1 Security
   - 7.2 Binary Compatibility.
   - 7.3 Reliability
   - 7.4 Maintainability.
   - 7.5. Portability
   - 7.6. Extensibility.
   - 7.7. Reusability.
   - 7.8 Application Compatibility.
   - 7.9. Resource Utilization
   - 7.10. Serviceability

8. Operational Scenarios
9. Preliminary Schedule
10. Preliminary Budget
11. Appendices
    11.1. Definitions, Acronyms, Abbreviations
    11.2. References.

→ Characteristics of good SRS -

(1) Correct
(2) Complete
(3) ~~Unambiguous~~ Unambiguous (unique interpretation)
(4) Consistent (no conflicts)

Three types of conflicts -
- logical and or temporal confict
- Characteristics conflicts of real world object
- Two different descriptions about the same real world object.

(5) Stability
(6) Verifiable
(7) Traceable ; Two types -

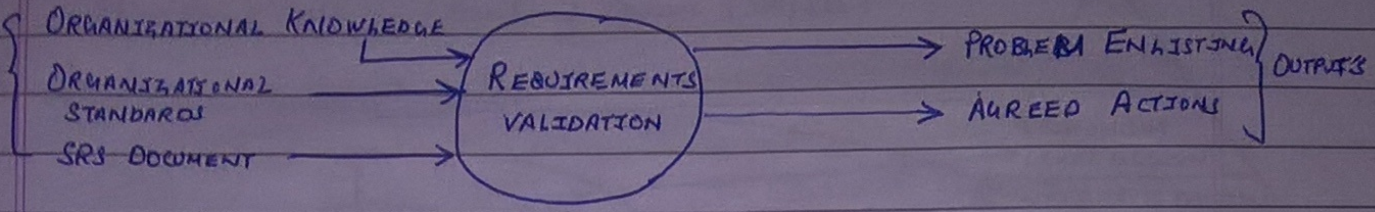
Forward Traceability - Each requirement is referred in the SRS document by its unique name or reference number.

Backward Traceability - If the reference to the requirement is mentioned in earlier document, then it is backward traceabili

⑥ Requirements Validation -

It is a technique in which the description in SRS docume is verified for system implementation. Checks made on SRS document a

(1) Completeness and consistency
(2) Conformance to standards
(3) Requirement conflicts
(4) Ambiguity in requirements.

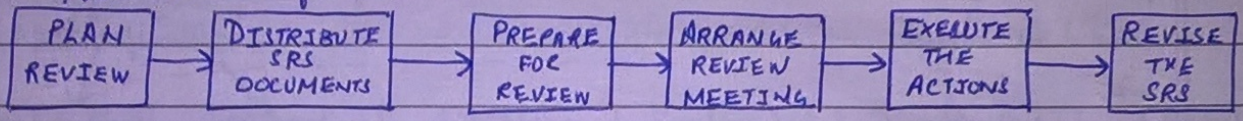



→ Requirement validation techniques -

(1) Requirement Review -

(1) read and analyse requirements
(2) identify the problems
(3) discuss the problems
(4) agree upon the actions to solve these problems

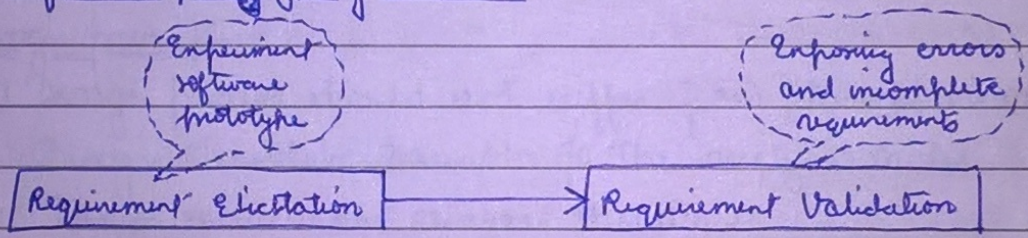
→ Process model for review -




(2) Software Prototyping -

Use of system prototype is to help customers and developers to understand the system requirements

Software prototyping activities -




(7) Traceability -

It is concerned with relationship between requirements, their sources and the system design. Using traceability the requirement finding becomes easy. Various types of traceability are -

(1) Source traceability - Stakeholders
(2) Requirements traceability - dependent requirements
(3) Design traceability - design

Traceability information is typically represented by a data structure called Traceability matrix.

R → weak relationship
D → dependent

| Requirement ID | A | B | C |
|---|---|---|---|
| A | S | D | R |
| B | D | | D |
| C | | | |