

Unit II - Requirements

Analysis and Specification

Software Requirements

Functional and Non-functional
User requirements, System
requirements, Software requirements
Document - Requirement Engineer
Process: Feasibility Studies,
Requirements elicitation and
analysis, requirements validation,
requirements management,
Classical analysis, Structure
System Analysis, Petri Nets
Data Dictionary.

SPECIFICATION

Introduction.

IEEE defines a requirement

as

1. A condition needed by a user to solve a problem.
2. A condition that must be met by a system to fulfil a contract.

Different types of Requirements

User Requirements

2. Requirements.

UNIT - II

REQUIREMENTS ANALYSIS AND SPECIFICATION

Introduction.

IEEE defines a requirement

as

1. A condition needed by a user to solve a problem.
2. A condition that must be met by a system to fulfil a contract.

Different types of Requirements

User Requirements

System Requirements.

Functional & Non-Functional

Requirements :-

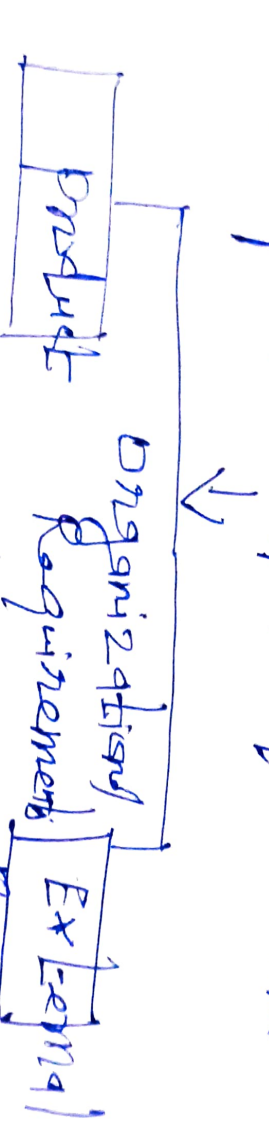
The system should be able to handle particular input and produce particular output and how the system should behave in a given situation.

Non-functional Requirements.

They represent the constraints placed on the system.

Domain Requirements.

These are from the application domain of the system. Non functional Requirements.



Explain Requirements

1. Legislative requirements.
2. Ethical requirements.

Useg Requirements

- should describe the functional and non functional requirements that they are understandable by system users who don't detailed technical knowledge.

a. Lack of clarity

b. Confusion.

c. Amalgamation.

Document.

1. Introduction.
2. General description
3. Specific Requirements.
4. Appendices.
5. Bibliography and Index.

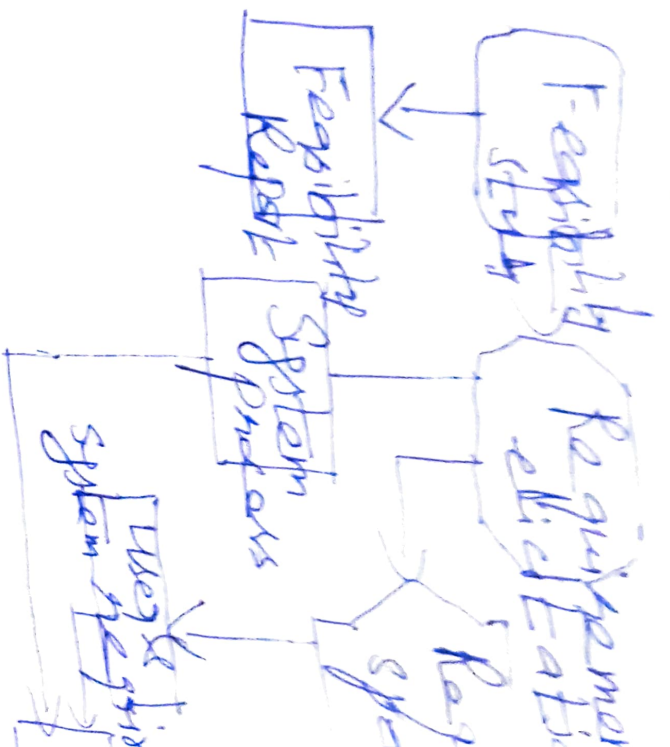
Requirements Engineering process

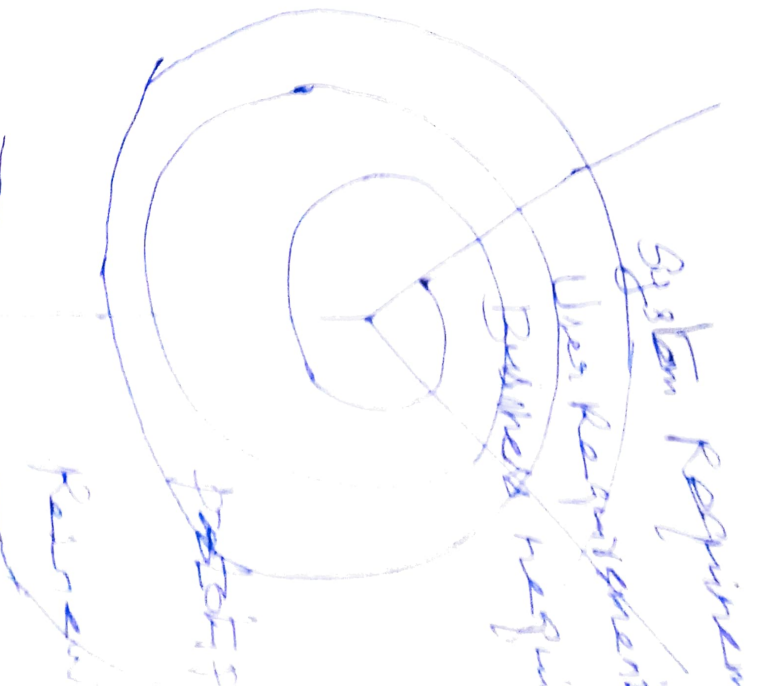
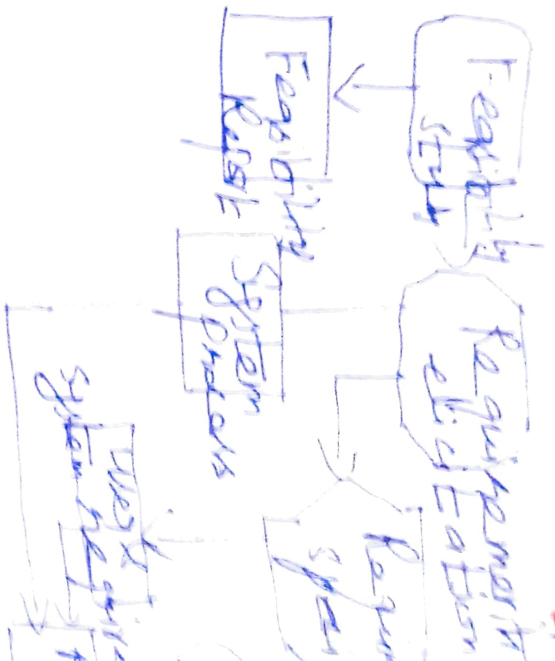
1. Requirements Elicitation
2. Analysis and

Negotiation.

3. Requirements Specification
4. Validation

5. Management.





Requirements Engineering

Business Requirements

System Requirements

User Requirements

System Analysis

System Design

System Implementation

Specification

The SRS requires
and specification for
on what the system
do not how the system
will be implemented.

Software the requirements
can be specified in the
following ways.

- ① Functional requirements
→ how the system should
react to particular inputs
behaves in particular

Validation makes sure that the requirements meet the customer needs

These checks include

Validity checks

" consistency "

" completeness "

" Verify ability "

~~Requirements Reviews~~

Reviews may check for

1. Verify ability

2. Consistency

prototyping

Test Case Generation

automated cross referencing.

Scenarios.

Reading.

Requirements Management

is a set of activities that help the project team to identify, control and track requirements.

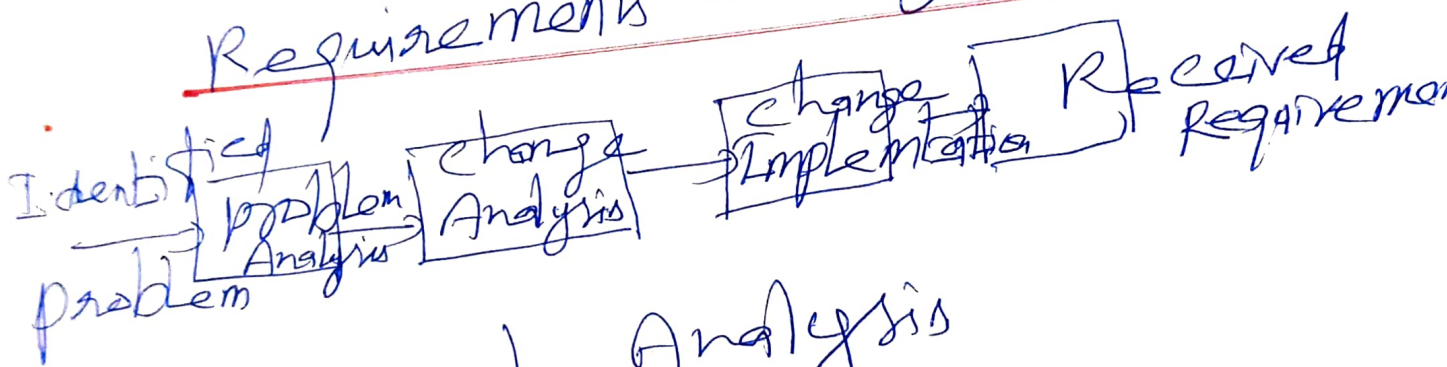
Requirements classes.

1. Enduring Requirements
2. Volatile "
3. Mutable "
4. Emergent "
5. Consequential "
6. Compatibility "

Requirements Management

1. Requirements identification
2. Change Management policies.
3. Traceability policies.

Requirements change Management.



classical Analysis

1. Introduction
2. products of classical Analysis ^{work} flow

3. The specification Documents

4. Software specification methods
5. Structured system Analysis

Analysts techniques are codes

- classical (structured) analysis

- Object-Oriented Analysis

Products of classical analysis

workflows

1. The specification document

- is a detailed design

2. Who: system will do

3. specify other documents

be free of some of the

3. Software Specifications

1. Informal methods

2. Semiformal

3. Formal

Structured System Analysis

1. Requirements

System Analysis

Notation below

Describe how a system processes in a shop

Minic

software example of

Study - An software item

Sallys software

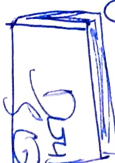
buy software \$ 10000 to

Suppliers and sell it to

Public the rate of

Step 1 Draw the data flow

Diagram [DFD] Four



Double square analysis

Flow of data

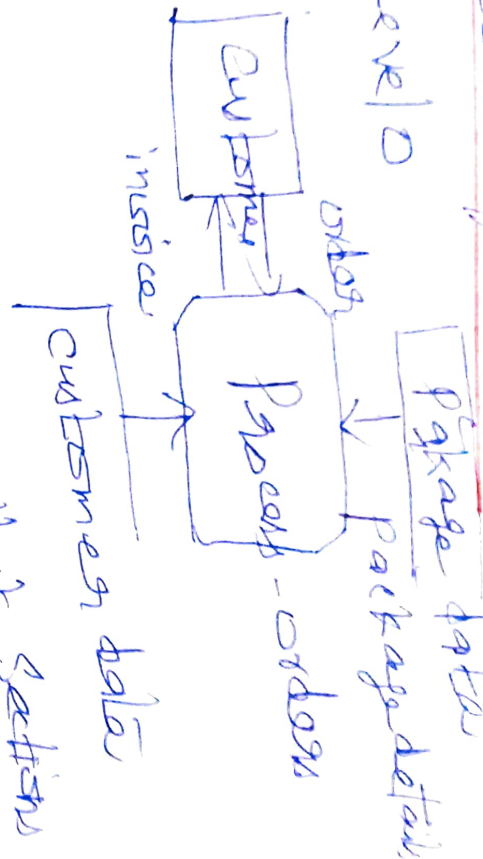


Process the transformation of data



Open and Refactor

DDD Level 0
Gallies.



Step 2: Decide what settings

to computerize and

[batch or online] high cost

Large

o batch.

Small volume, inhouse

microcomputer.

- On line

Step 3: Determine the details

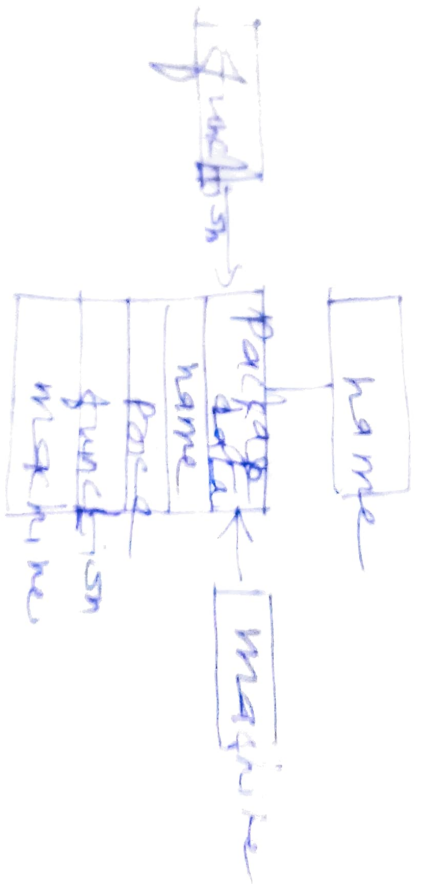
of data flows.

Refine Each Stepwise

Step 4: Define parameters



Step 5: Define data, score



Step 6: Define Physical the sources.

For each file specify file name

Step 7: Determine Input-Data
Specification.

Specify: Input forms,
Input Screens

Step 8: Determine Sizing.

- Volume of inputs
- mass storage
- size of each file
- hardware requirements.

Steps: Determine requirements.

Overall:
The method of Gilbre /
De Mores / has resulted
major improvements in the
software industry.

structured

The way can program have
different types of operations

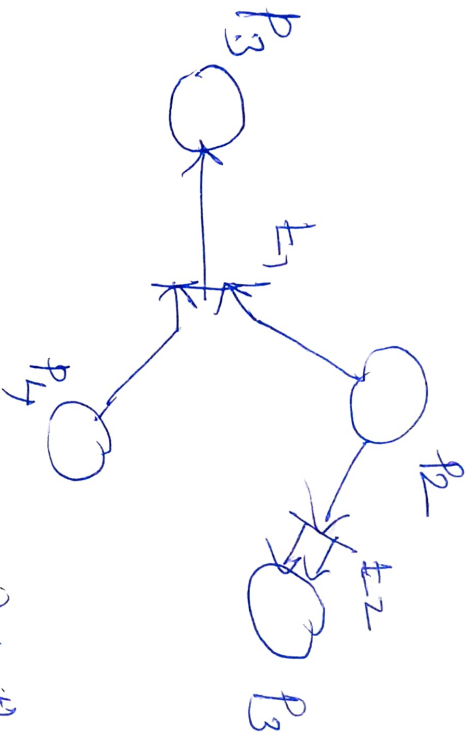
1. Update investment data
2. print a list of investments
3. print a report.

peer nets.

A powerful technique
for specifying systems that have
potential problems with inter
relationships

- Synchronization problems
- Race conditions
- Dead lock

four parts.



A Petri net is considered if

four parts

• A set of places.

• A set of transitions T

• An input function I

• An output function O

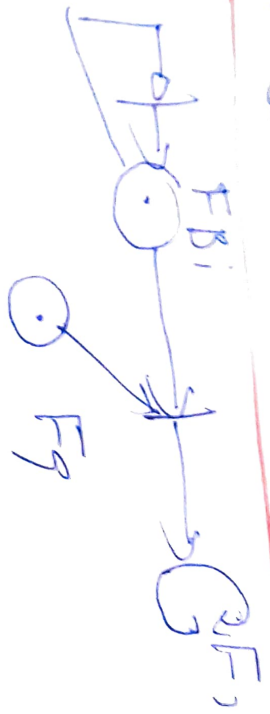
Output functions:

$$O(t_1) = \{P_1\}$$

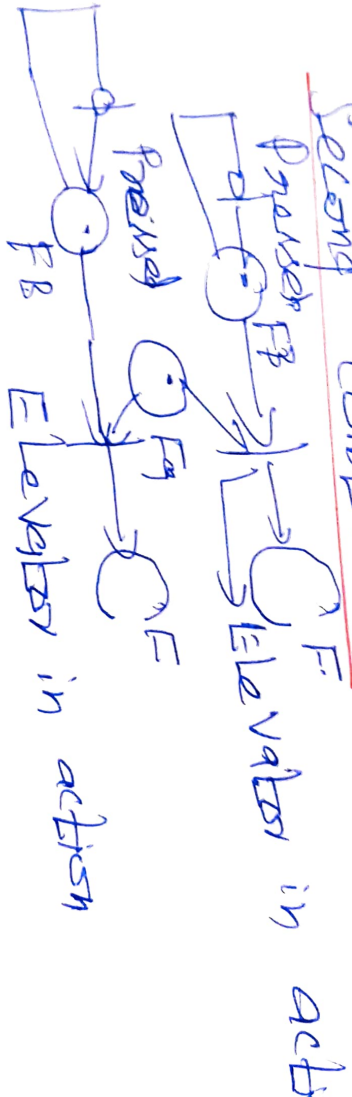
$$O(t_2) = \{P_2, P_3\}$$

Let's Notes : the
problem
are stud

First constraint



Second constraint



Third constraint

- If an elevator has no requests it remains at its current floor when its door closed.
- Petri nets can also used

Data Flow Diagram

Data Nets

A powerful technique for specifying systems that have potential problems with interrelations.

Data Dictionary

A data dictionary is an organized listing of all data elements that are relevant to the system.

As both the user and system analyst ~~will~~ have a

common understanding in terms, outputs components of stores

1. Name of
2. Aliases
3. Description.
4. Related data items
5. Range of value.
6. Data structure definition.
7. Where used
8. Supplementary information.

Advantages

1. Mechanism for name management.
2. It serves as a

analysis design implementation
and evolution.

3. The data dictionary software
might be integrated with
other tools so that
dictionary creation is
partially automated.

4. Design the software
and test cases.

XXXXXXXX