# Requirement Analysis

Observation of existing systems

Studying existing procedures

Discussion with the customer and the end-users

Having questionnaires to understand user requirements

 Conducting discussion with domain experts to understand the system

Requirement gathering

Requirement gathering for automating system · Analyst can easily obtain

✓ Input and output formats

✓ Accurate details of the operational procedures

Analyzing the gathered requirement

- Understand the user requirements,
- Detect and remove inconsistencies, ambiguities, and incompleteness

Incompleteness and inconsistencies

Resolved through further discussions with the endusers and the customers

# **Contradicting Requirement**



One requirement conflicts with another requirement.



Tom have you come across any contradicting requirements? Can you give an example?

#### Sure!

When pouring hot filling into chocolate candy shells,

- the filling should be hot enough to pour quickly
- but it should also be cold enough to prevent melting of the chocolate.

## **Incomplete Requirement**

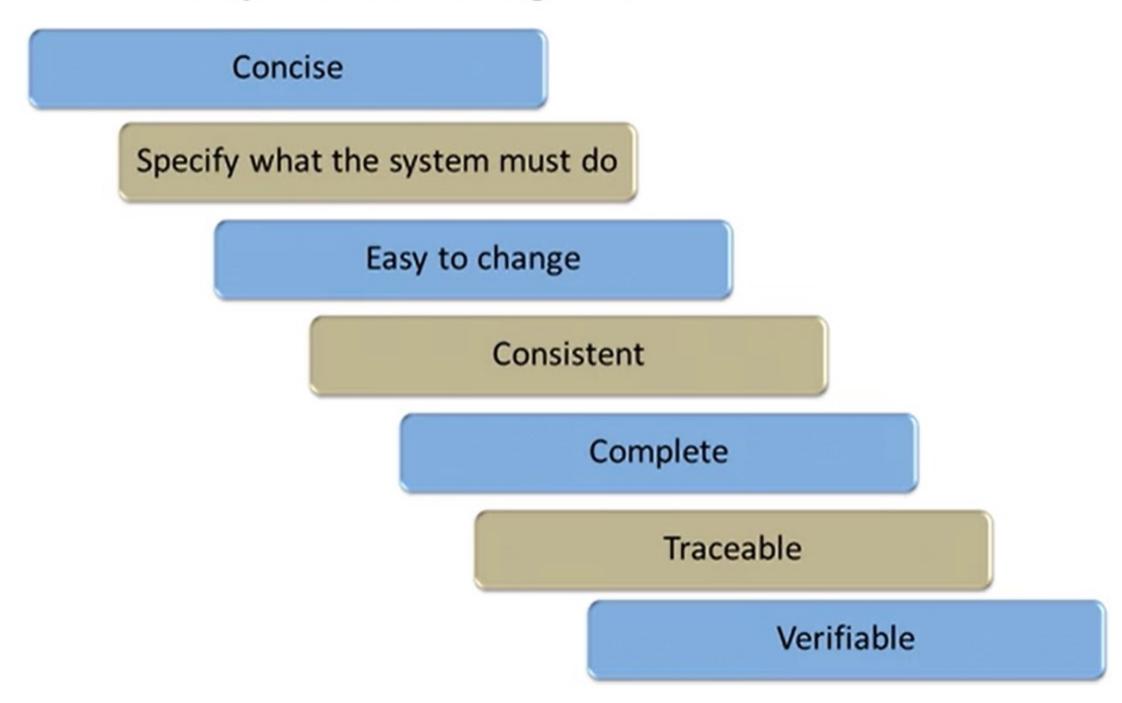
Some requirements have been omitted:

Due to oversight.

#### Example:

- A solar heater specifies what it will do on a 'sunny day' but it fails to specify what it will do on a 'rainy day'
- The analyst has not recorded when temperature falls below 90 degrees.
  - · heater should be turned ON
  - water shower turned OFF.

# **Properties of a good SRS**



### **SRS Document**

SRS document normally contains three important parts

SRS **Functional** requirements Nonfunctional requirements Constraints on the system.

# **Functional Requirement**

### Functional requirements describe

- A set of high-level requirements
- Each high-level requirement
- Takes in some data from the user
- Outputs some data to the user
- Might consist of a set of identifiable functions which process the input

# **Functional and Non Functional Requirement**



Tom, please list few functional Requirements for a Matrimonial website.

Tom can you list few non functional requirements?

You are right!

Security and Performance

are also non functional

roquiromonto

Register User

Match Horoscope

Update User Details

Members Online

· Search by mother

tongue

- ✓ Maintainability
  - ✓ Portability
  - ✓ Usability
  - ✓ Reliability
  - ✓ Robustness



### Constraints



Constraints describe things that the system should or should not do.

- · For example,
  - Standards compliance
  - How quickly the system can produce results so that it does not overload the other system to which it supplies data, etc.
  - Hardware to be used
  - Operating system
  - DBMS to be used
  - Capabilities of I/O devices
  - Data representation

### **SRS Document Structure**



1. Introduction	. 1
1.1 Purpose	. 1
1.2 Document Conventions	. 1
1.3 Intended Audience and Reading Suggestions	. 1
1.4 Project Scope	. 1
1.5 References	. 1
2. Overall Description	
2.1 Product Perspective	: 2
2.2 Product Features	2
2.2 Product Features	. 2
2.4 Operating Environment	2
2.4 Operating Environment 2.5 Design and Implementation Constraints	. 2
2.6 User Documentation	. 2
2.7 Assumptions and Dependencies	. 3
3. System Features	. 3
3.1 System Feature 1	. 3
3.2 System Feature 2 (and so on)	. 4
4. External Interface Requirements	4
4.1 User Interfaces	- 4
4.2 Hardware Interfaces	4
4.3 Software Interfaces	-4
4.4 Communications Interfaces	4
5. Other Nonfunctional Requirements	5
5.1 Performance Requirements.	. 5
5.2 Safety Requirements	. 3
5.2 Safety Requirements 5.3 Security Requirements	. 5
5.4 Software Quality Attributes	. 5
6. Other Requirements	5
Appendix A: Glossary	
Appendix B: Analysis Models	
Appendix C: Issues List	. 6

### **ERD**



#### Entity

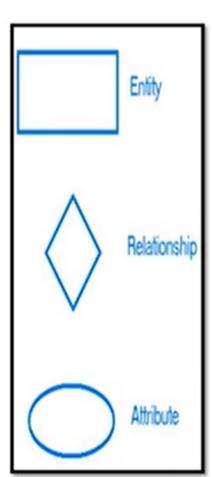
 An entity is a business object that represents a group, or a category of data.

#### **Attribute**

Properties of an entity.

#### Relationship

· specifies the relations among entities







Relationship specifies association between two entities.

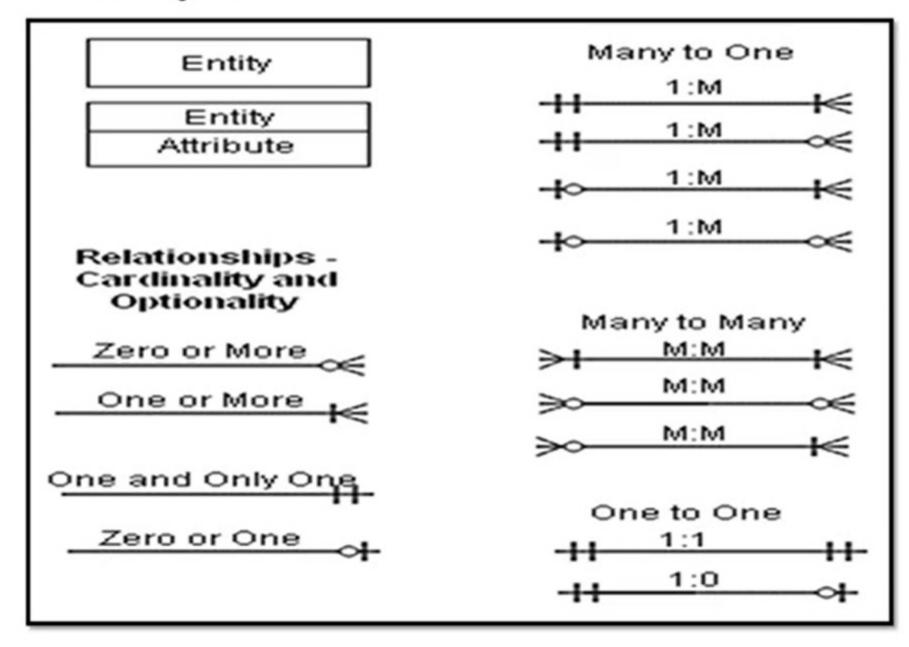
### Cardinality

- One instance of an entity maps to how many instance of other entity
- Many-to-Many Relationships
- One-to-Many Relationships
- One-to-One Relationships
- Recursive Relationships

### Optionality

- Is the relationship mandatory or optionality
- Mandatory Relationships
- Optional Relationships

### **ERD Example**



#### Why is Requirement Analysis difficult?

#### Different "worlds"

- Bridging the gap between the client and the software developer
- Knowing what should be done VS knowing what to let a computer do

#### Users/stakeholders are not a uniform group

- conflict between cost and usability / performance / features
- conflicting demands from different departments

Getting the good (ideal) system

Vs

possibility of building it well