

Tree 3

Date

No

Requirement Engineering

→ Addresses two main problems

- > What do we want to build?
- > How do we write this down?

What is RE?

Requirement Engineering

Requirements engineering is a process of

establishing

> The functions and attributes

→ that a customer requires from a system.

> The constraints

→ Under which it operates and it is developed.

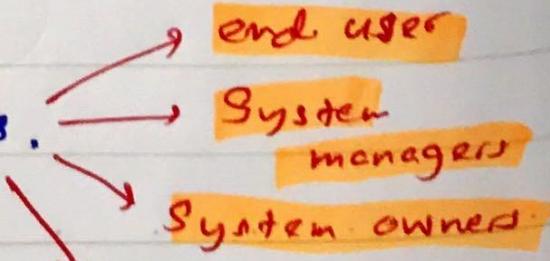
Why RE?

- > Trouble in understanding what customer really wants
- > Record requirements in a disorganized manner.
- > Spend far too little time verifying what we do record.
- > Fail to establish a solid foundation for the system or software that the user wants built.

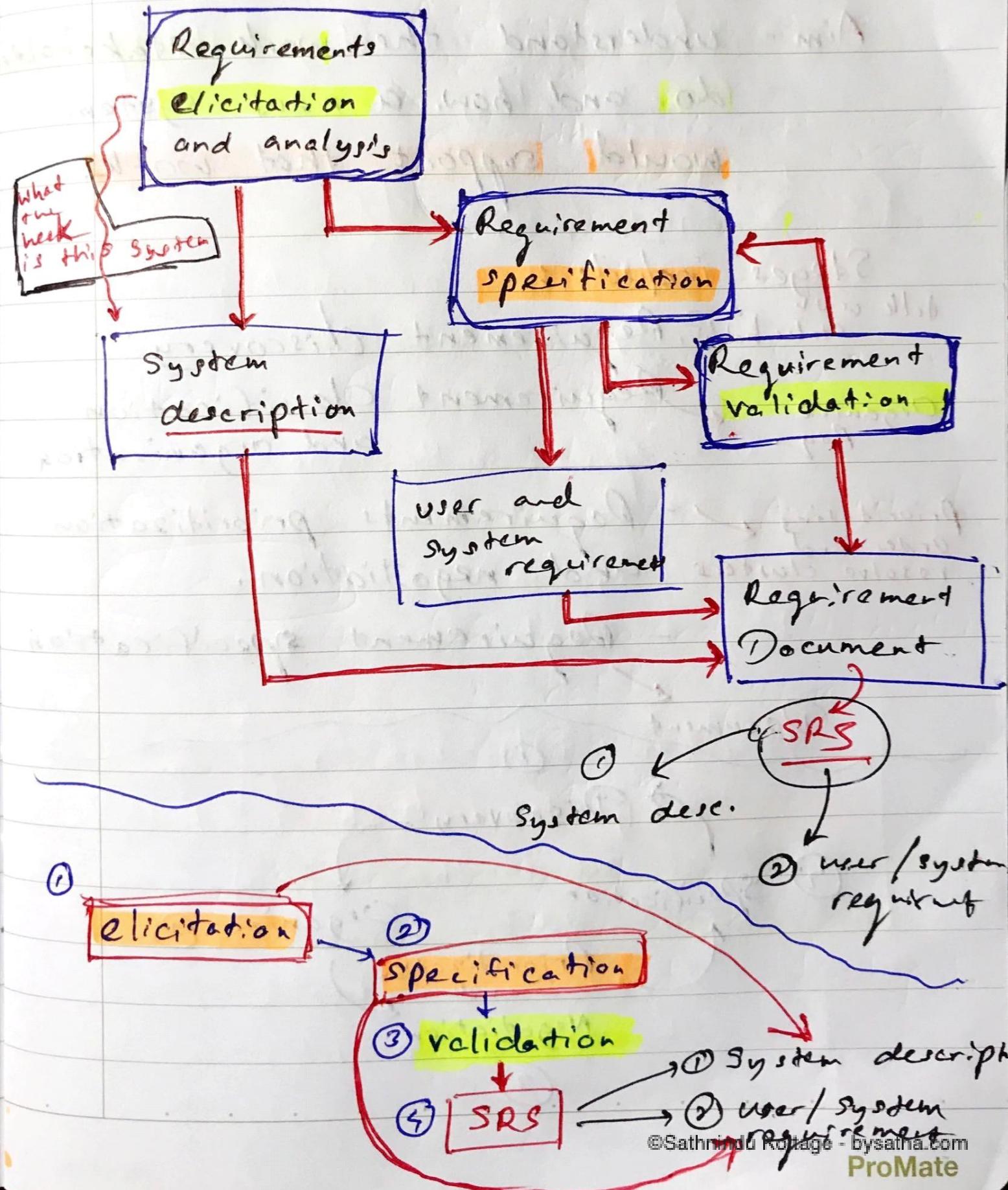
System stakeholders

- > Any person / organization who is affected by the system in some way and so who has a legitimate interest.

Stakeholder types.



Requirement Engineering Process.



① Requirement elicitation

Aim - understand the work stakeholders do and how a new system would support that work.

Stages include:

^{talk with} Stakeholders → Requirement discovery

Organize Req. → Requirement classification and organization

Prioritising ← Requirements prioritization
order and resolve clusters and negotiation.

→ Requirement specification

Document

①

Discovery

④

Specification

Organize ②

Prioritization and Negotiation

③

① Requirements Discovery

> The process of gathering information about the required and existing systems and distilling the user and system requirements from this information.

Interviewing

Observation/
Ethnography

closed or open

② Requirement Classification

> Functional requirements

Statements of services the

system should provide,

how the system should

react to particular inputs

and how the system should

behave in particular situations

> Non-functional requirements

Requirements that are not directly concerned with specific functionality.

> constraints

Functional Requirements

① Describe functionality or g. system services.

② Depend on the type of software, expected users and the type of system where the software is used.

③ Functional user requirements may be high-level statements of what the system should do but functional system requirements should describe the system services in detail.

Non-functional requirements

- ① Often apply to the system as a whole rather than individual features or services
- ② These define system properties.
- ③ Are also called **Quality attributes**.
- ④ Non-functional requirements may be more critical than functional requirements. If these are not met, the system may be useless.

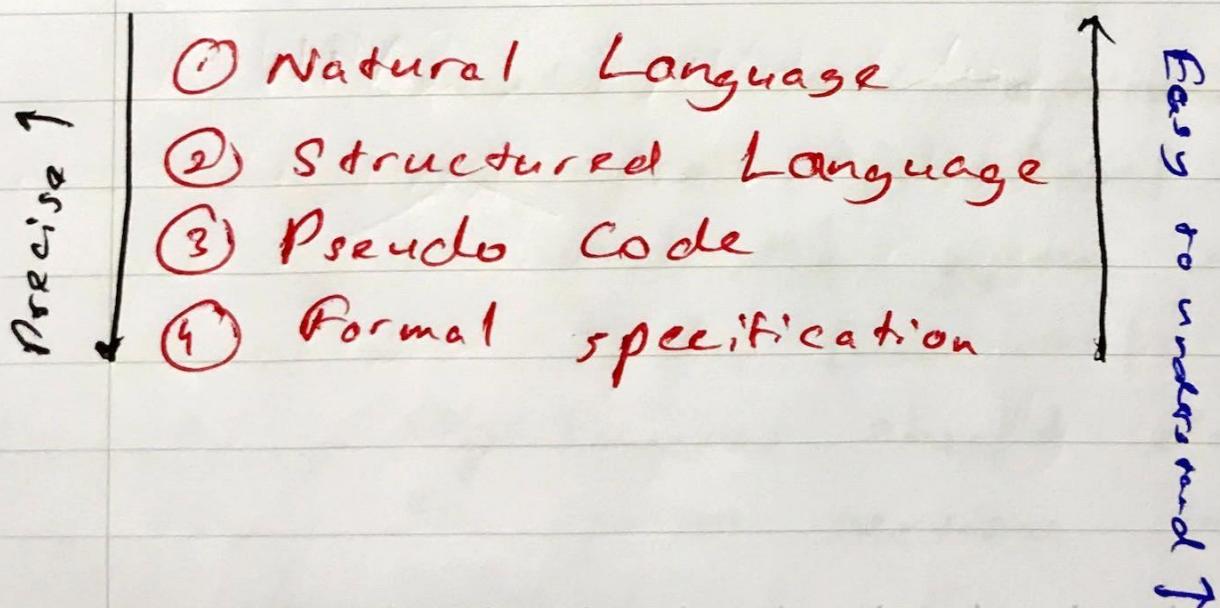
Requirement imprecision

> Uncertain Requirements
Changes over time
ex: LiveLanka projects

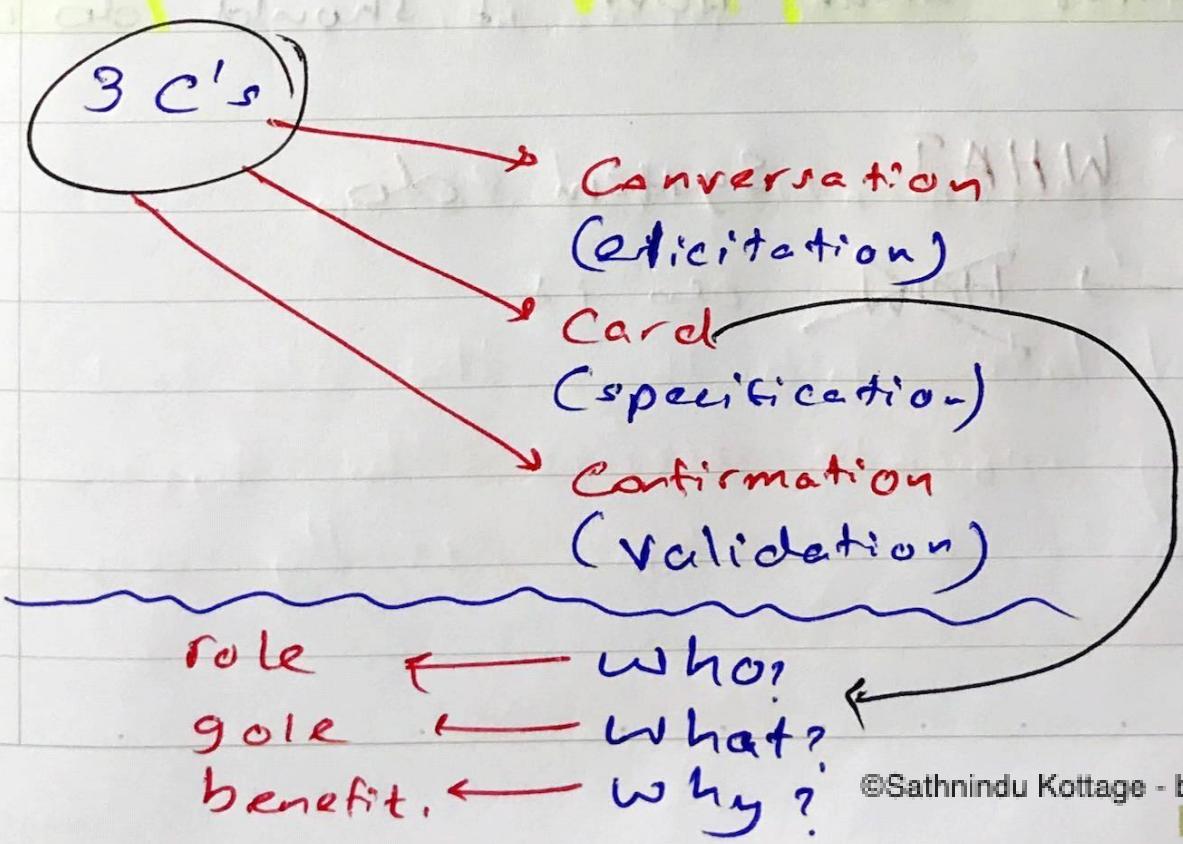
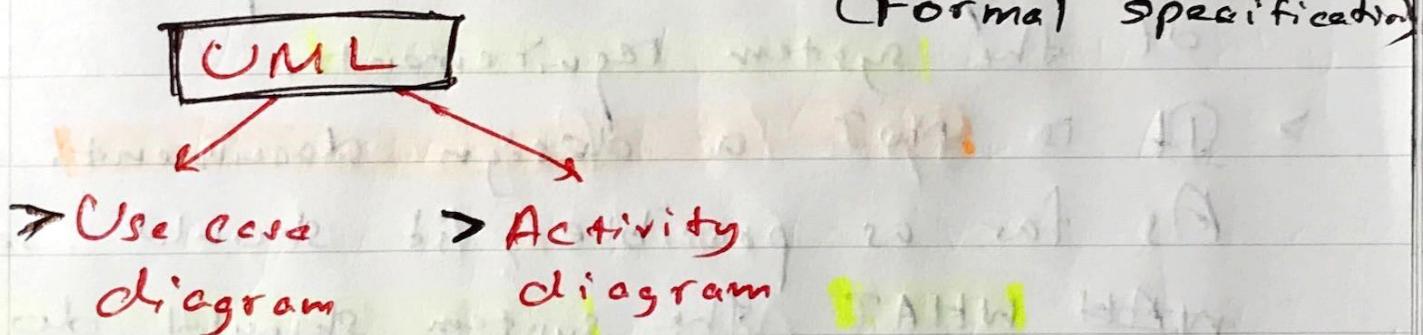
> Ambiguous Requirements
ex: Rutherford projects

> Incomplete Requirements

System requirements Specification



- Natural language
- Structured natural language (Forms)
- Design description language (~~code~~)
(pseudo code)
- Graphical notations (UML)
- Mathematical specification
(Finite-state machines or sets)
(Formal specification)



SRS

- > Software Requirements Specification
- > The software requirements document is the official statement what is required of the system developers.
- > Should include both a definition of user requirements and a specification of the system requirements.
- > It is NOT a design document.
As far as possible, it should set of ~~not~~ **WHAT** the system should do rather than **HOW** it should do it.

WHAT System do

NOT → ~~How~~ do it

Users of SRS

System customers } → Read & check

Managers → Plan

System engineers } → Develop

System test eng. } → Validation test

System maintenance engineer } → Understand the relationship

- SRS
- ① Should correctly define all the software requirements.
 - ② Should not describe any design or implementation details.
 - ③ Should not impose additional constraints on the software.

SRS Template

> Introduction

- Purpose, Scope, Overview

> General Description

- Product Perspective,
- User Characteristics

> Specific Requirements

- Functional Requirements
- Non-functional
- Constraints

Requirements Validation

- > Validate whether the elicited requirements define the system that the customer really wants.

- > Requirements error costs are high so validation is very important
- > Requirements checking
 - Validity
 - Precise requirements
 - Realism
 - Verifiability
- > Requirements reviews
 - Systematic manual analysis of the requirements.
- > Prototyping
 - Using an executable model of the system to check requirements.
- > Test - case Generation
 - Developing tests for requirements to check testability
 - ~~with the document later~~