Software Engineering Principles

Requirements analysis & specification

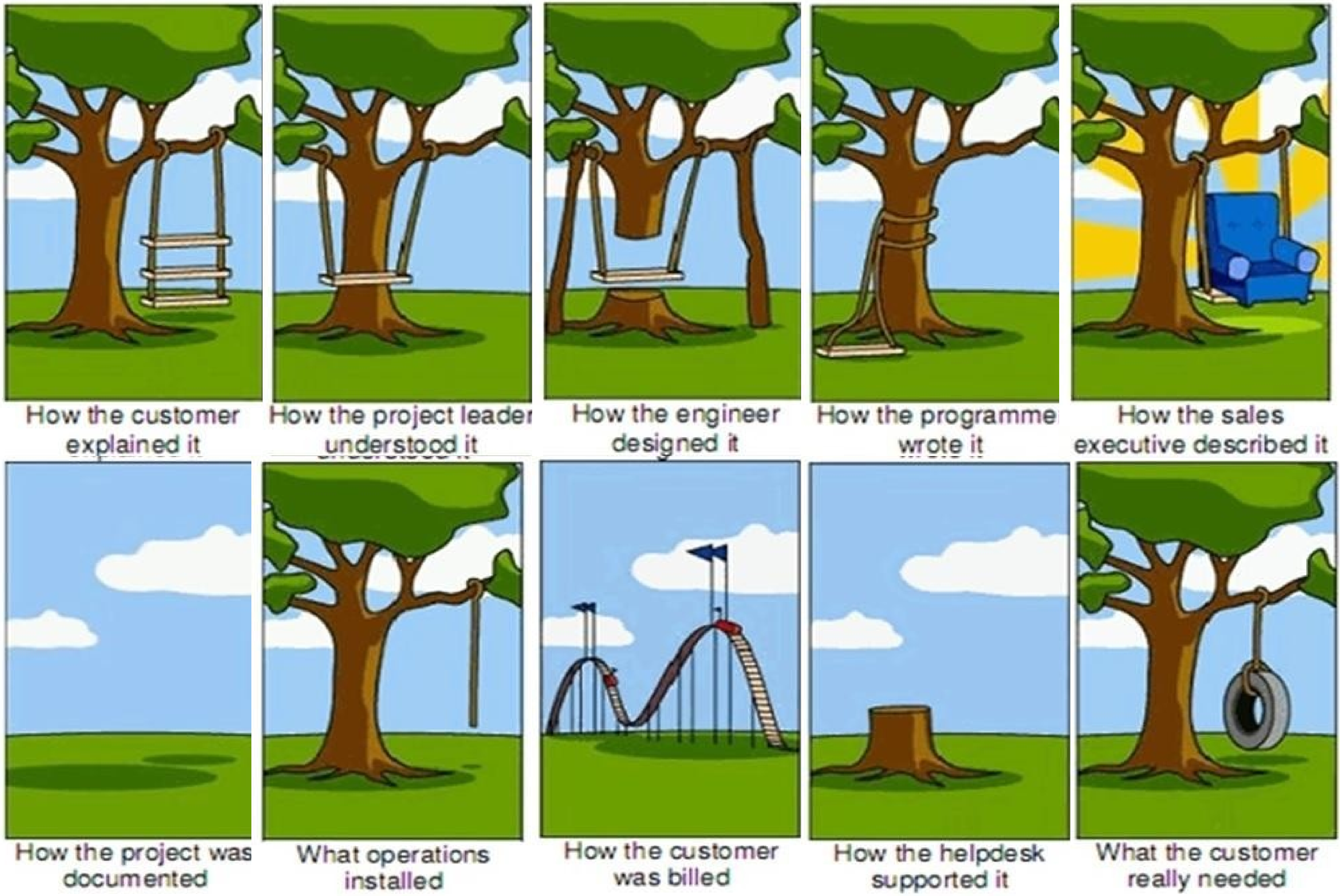
Acknowledgements

- Slides of Prof. Rajib Mall, IIT, KGP

Why Study Software Engineering?

- Many projects have failed because
  - they started to develop without adequately determining whether they are building what the customer really wanted.
- Customer Requirement

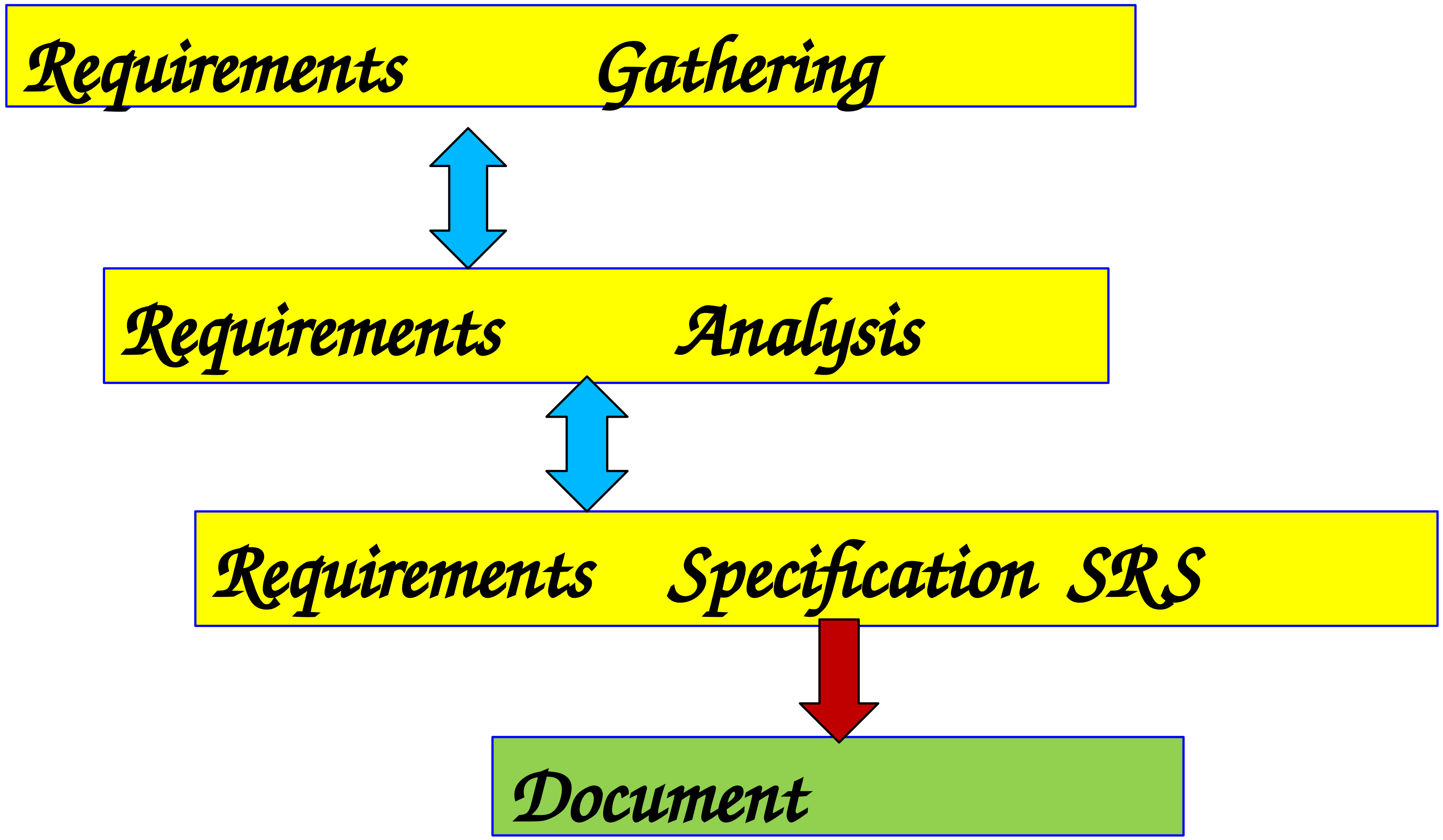
Typical project scenario…


How the customer explained it
How the project leader understood it
How the engineer designed it
How the programme wrote it
How the sales executive described it
How the project was documented
What operations installed
How the customer was billed
How the helpdesk supported it
What the customer really needed

Requirements

- A Requirement is a capability or condition required from the system.
- What is involved in RAS?
  - Determine what is expected by the client from the system. (Gather and Analyze)
  - Document those in a form that is clear to the client as well as to the development team members. (Document)

Activities in RAS



Requirements engineering

- The process of establishing the services that
  - the customer requires from a system and
  - the constraints under which it operates and is developed.
- The requirements themselves are the descriptions of
  - the system services and
  - constraints that are generated during the requirements engineering process.

Requirements Analysis and Specification

- Requirements Gathering:
  - Fully understand the user requirements.
- Requirements Analysis:
  - Remove inconsistencies, anomalies, etc. from requirements.
- Requirements Specification:
  - Document requirements properly in an SRS document.

Need for SRS…

- Good SRS reduces development cost
  - Req. errors are expensive to fix later
  - Req. changes cost a lot (typically 40% changes)
  - Good SRS can minimize changes and errors
  - Substantial savings --- effort spent during req. saves multiple times that effort

Uses of SRS Document

- Establishes the basis for agreement between the customers and the suppliers
- Forms the starting point for development.
- Provide a basis for estimating costs and schedules.
- Provide a baseline for validation and verification.
- Facilitates transfer.
- Serves as a basis for enhancement.
- The SRS can serve as the basis for writing User Manual for the software:
  - User Manual: Describes the functionality from the perspective of a user --- An important document for users.
  - Typically also describes how to carry out the required tasks with examples.

SRS Document: Stakeholders

- SRS intended for a diverse audience:
  - Customers and users for validation, contract, ...
  - Systems (requirements) analysts
  - Developers, programmers to implement the system
  - Testers to check that the requirements have been met
  - Project Managers to measure and control the project
- Different levels of detail and formality is needed for each audience
- Different templates for requirements specifications:
  - Often variations of IEEE 830

Types of Requirements

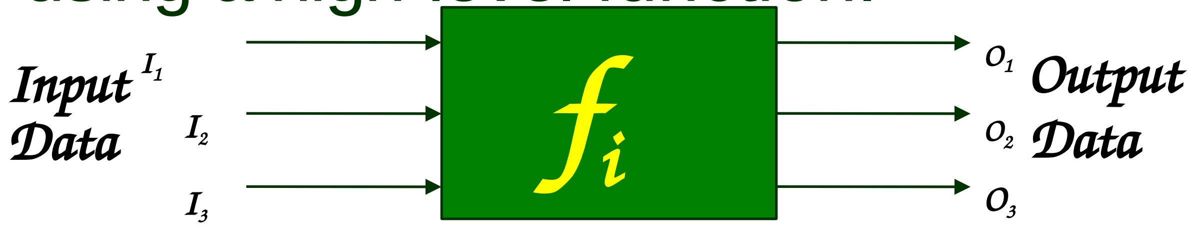
- Functional requirements
  - Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.
- Non-functional requirements
  - Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process etc.
  - Often apply to the system as a whole rather than individual features or services.

Functional Requirements

- Descriptions of data to be entered into the system
- Descriptions of operations performed by each screen
- Descriptions of work-flows performed by the system
- Descriptions of system reports or other outputs
- Who can enter the data into the system?
- How the system meets applicable regulatory requirements

Functional Requirements contd.

- The functional requirements discusses the functionalities required from the system.
- The system is considered to perform a set of high- level functions $\{f_i\}$
- Each function f_i of the system can be considered as a transformation of a set of input data (I_i) to the corresponding set of output data (O_i)
- The user can get some meaningful piece of work done using a high-level function.



Example Functional Requirements - I

- Interface requirements
  - Field 1 accepts numeric data entry.
  - Field 2 only accepts dates before the current date.
  - Screen 1 can print on-screen data to the printer.
- Business Requirements
  - Data must be entered before a request can be approved.
  - Clicking the Approve button moves the request to the Approval Work flow
- Regulatory/Compliance Requirements
  - The database will have a functional audit trail.
  - The system will limit access to authorized users.
  - The spreadsheet can secure data with electronic signatures.

Example Functional Requirements - II

- Library system - F1: Search Book function
  - Input: an author’s name
  - Output: details of the author’s books and the location of these books in the library
- ATM (Cash Withdraw)- R1: withdraw cash
  - Description: The withdraw cash function determines the type of account that the user has and the account number from which the user wishes to withdraw cash.
  - It checks the balance to determine whether the requested amount is available in the account.
  - If enough balance is available, it outputs the required
    cash, otherwise it generates an error message.

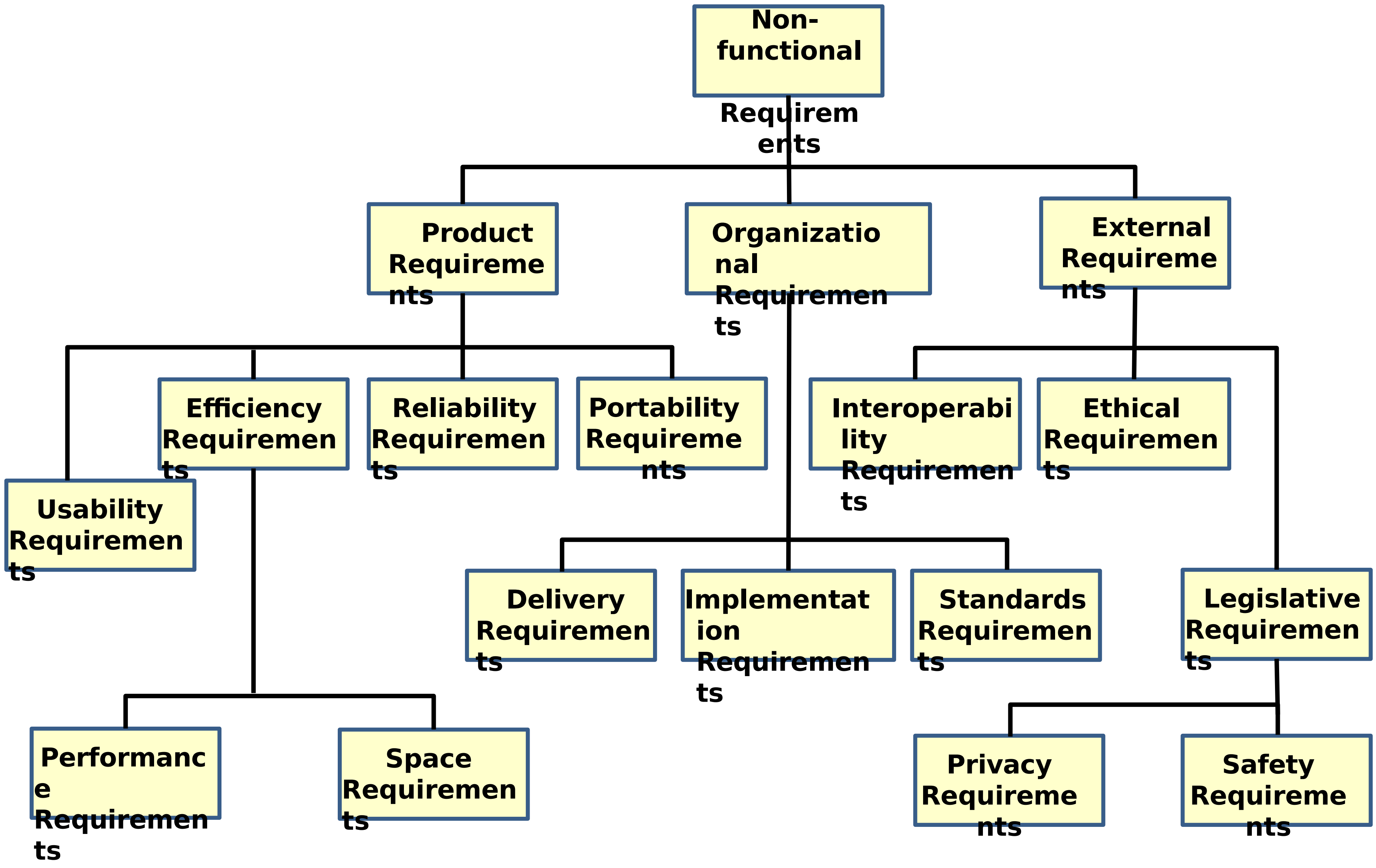
Example Functional Requirements - III

- ATM (Cash Withdraw)- R1: withdraw cash
  - R1.1: select withdraw amount option
    - Input: "*withdraw amount*" option,
    - Output: user prompted to enter the account type
  - R1.2: select account type
    - Input: user option,
    - Output: prompt to enter amount
  - R1.3: get required amount
    - Input: amount to be withdrawn in integer values greater than 100 and less than10,000 in multiples of 100.
    - Output: The requested cash and printed transaction statement.

Non-functional Requirements - I

- Characteristics of the system which can not be expressed as functions
  - Maintainability, Portability, Usability, Security, Safety, Reliability, Performance, etc.
- Example: How fast can the system produce results?
  - So that it does not overload another system to which it supplies data, etc.
  - Needs to be measurable (verifiability)
    - e.g. response time should be less than 1sec, 90% of the time

Non-functional Requirements - II



Non-functional Requirements III

- Constraints are NFR
  - Hardware to be used,
  - Operating system
  - DBMS to be used
  - Capabilities of I/O devices
  - Standards compliance
  - Data representations by the interfaced system
- Project management issues (costs, time, schedule) are often considered as non-functional requirements

Importance of Nonfunctional Req.

- Non-functional (product) requirements play an important role for critical systems.
  - Systems whose 'failure' causes significant economic, physical or human damage to organizations or people.
- There are three principal types of critical system
  - Business critical systems : Failure leads to significant economic damage.
    - customer accounting system in a bank
  - Mission critical systems : Failure leads to the abortion of a mission.
    - navigational system for a spacecraft
  - Safety critical systems: Failure endangers human life.
    - a controller of a nuclear plant

IEEE 830-1998 Standard for SRS - I

- Title
- Table of Contents
- 1. Introduction
- 2. Overall Description
- 3. Specific Requirements
- 4.Change Management Process
- 5. Document Approval
- Appendices
- Index

IEEE 830-1998 Standard: Introduction

- 1.1 Purpose
  - Describe purpose of this SRS
  - Describe intended audience
- 1.2 Scope
  - Identify the software product
  - Enumerate what the system will and will not do
  - Describe user classes and benefits for each
- 1.3 Definitions. Acronyms, and Abbreviations
  - Define the vocabulary of the SRS (may reference appendix)
- 1.4 References
  - List all referenced documents including sources (e.g., Use Case Model and Problem Statement; Experts in the field)
- 1.5 Overview
  - Describe the content of the rest of the SRS
  - Describe how the SRS is organized

IEEE 830-1998 : Overall Description

- 2.1 Product Perspective
  - Present the business case and operational concept of the system
  - Describe how the proposed system fits into the business context
  - Describe external interfaces: system, user, hardware, software, communication
  - Describe constraints: memory, operational, site adaptation
- 2.2 Product Functions
  - Summarize the major functional capabilities
  - Include the Use Case Diagram and supporting narrative (identify actors and use cases)
  - Include Data Flow Diagram if appropriate
- 2.3 User Characteristics
  - Describe and justify technical skills and capabilities of each user class
- 2.4 Constraints
  - Describe other constraints that will limit developer's options; e.g., regulatory policies; target platform, database, network software and protocols, development standards requirements
- 2.5 Assumptions and Dependencies

IEEE 830-1998 : Specific Requirements

- 3.1 External Interfaces
  - Detail all inputs and outputs
  - Examples: GUI screens, file formats
- 3.2 Functional Requirements
  - Include detailed specifications of all the functional requirements
- 2.3 Non-Functional Requirements
  - Describes all non-functional requirements that can't be expressed as a function.
    - Characteristics of the system which can not be expressed as functions.
    - e.g. performance requirements, database requirements, design constraints, quality attributes, . . .

Properties of a good SRS document

- Concise.
- Structured.
- Black-box view.
- Conceptual integrity.
- Response to undesired events.
- Verifiable.

Thank You