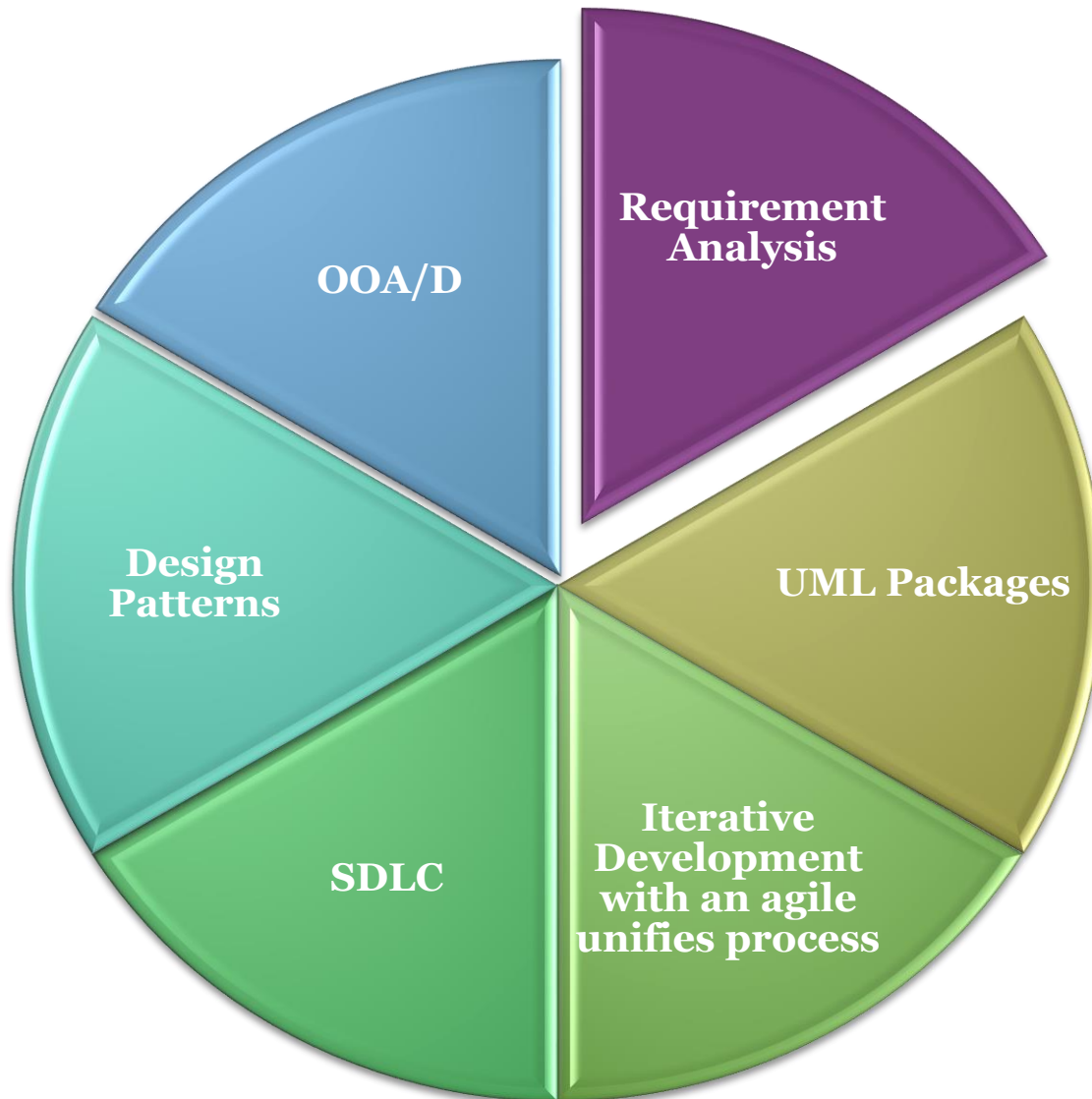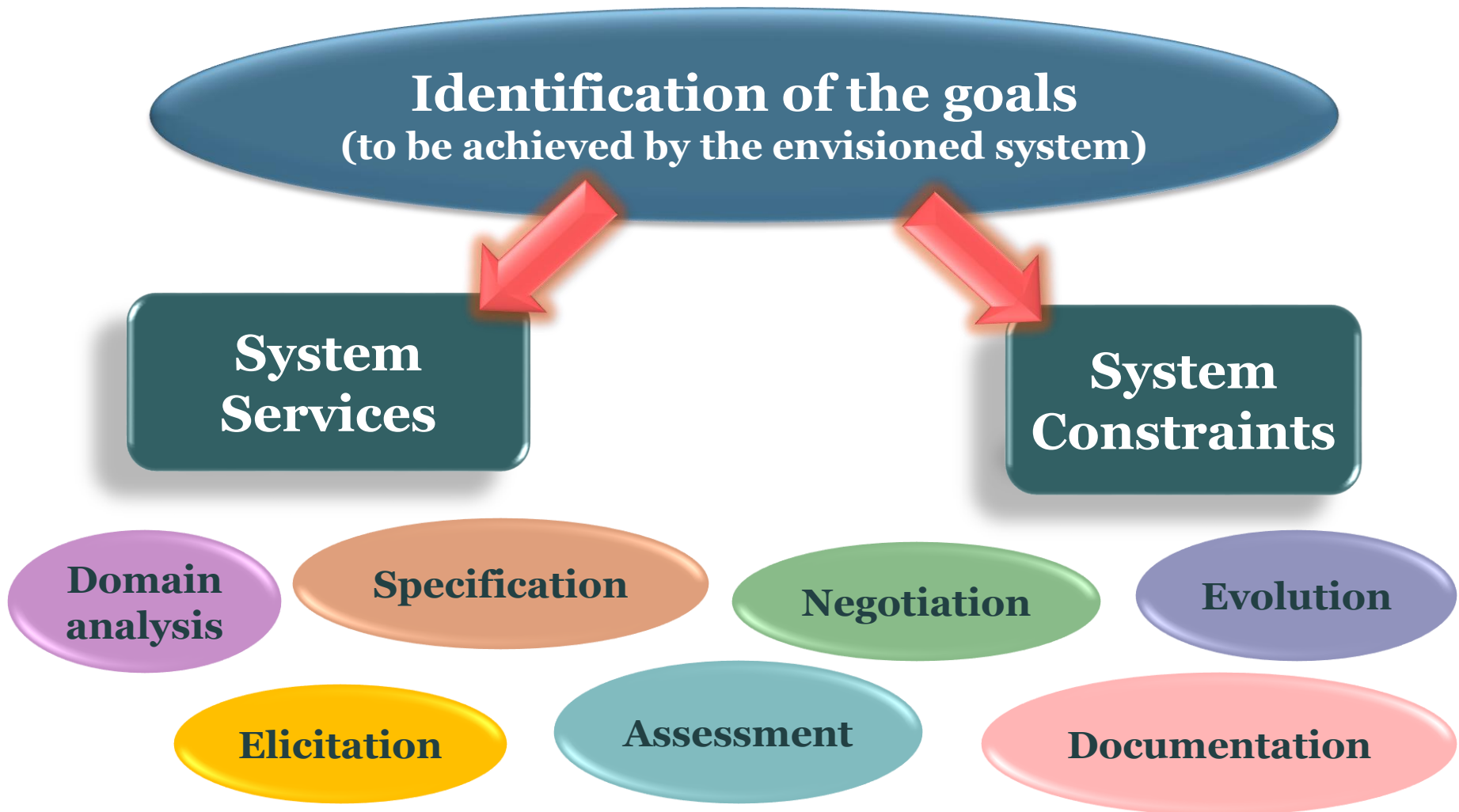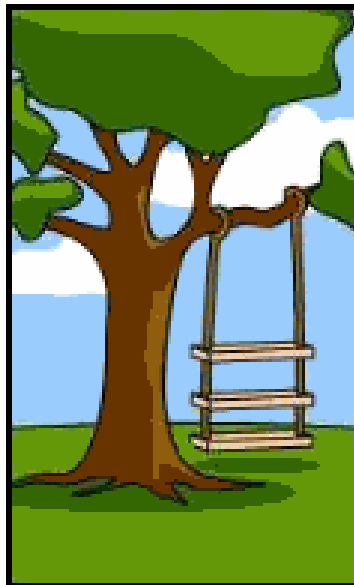# Requirements Engineering

Lecture # 3

Agenda

- Functional and Non-Functional Requirement
- Requirements Engineering Processes
- Requirements Elicitation
- Requirements Specification
- Requirements Validation

What is Requirements Engineering?

Identification of the goals
(to be achieved by the envisioned system)

System Services

System Constraints

Domain analysis

Specification

Negotiation

Evolution

Elicitation

Assessment

Documentation
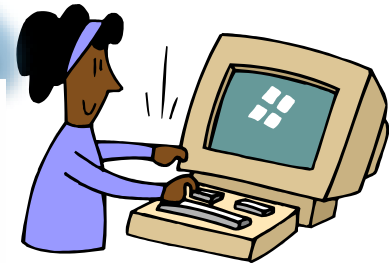
Where are Requirements Engineers standing?

Where are Requirements Engineers standing?

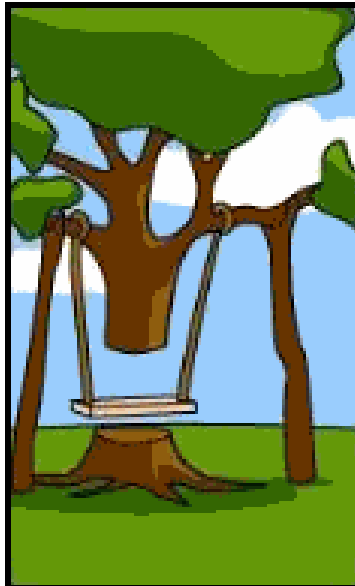How the Analyst designed it
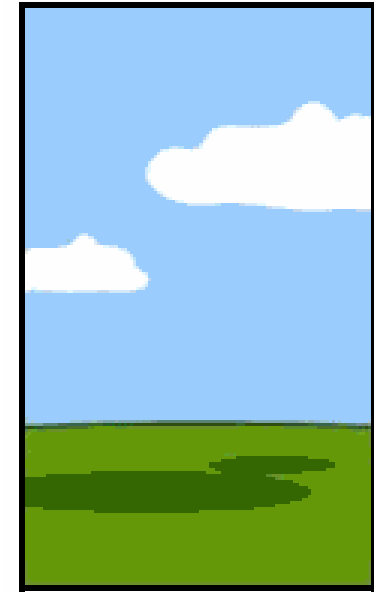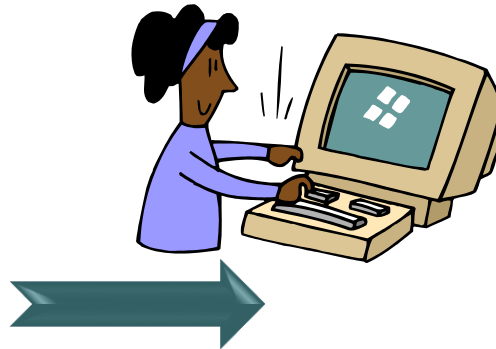
How the project was documented

How the customer explained it

How the Project Leader understood it

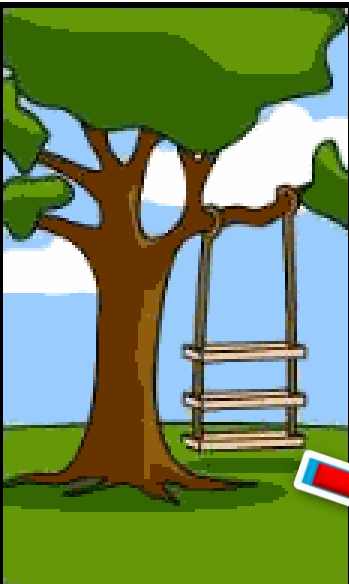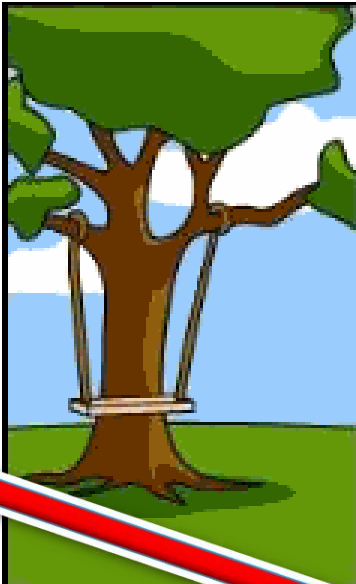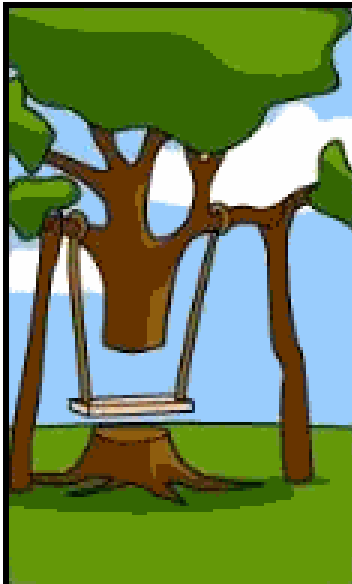How the Analyst designed it
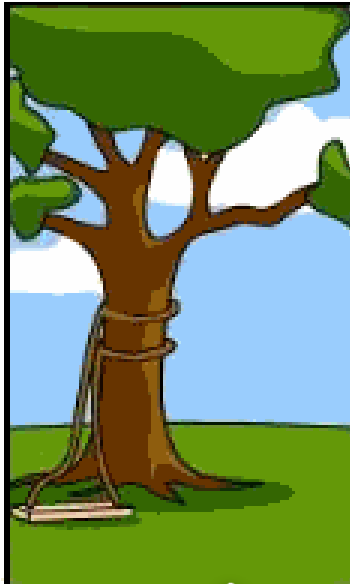
How the Programmer wrote it

How the Business Consultant described it
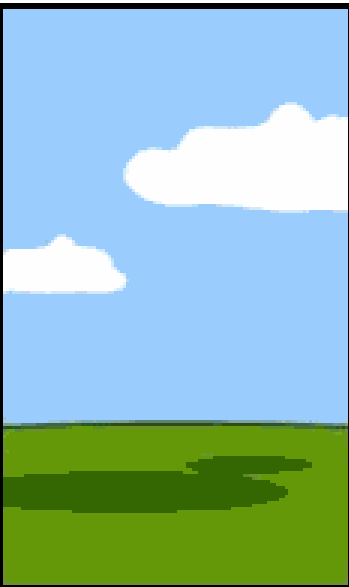
How the project was documented
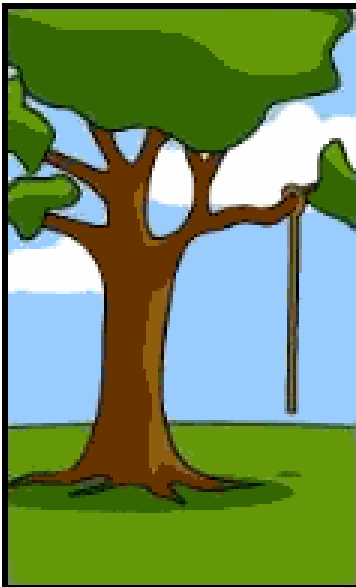
What operations installed
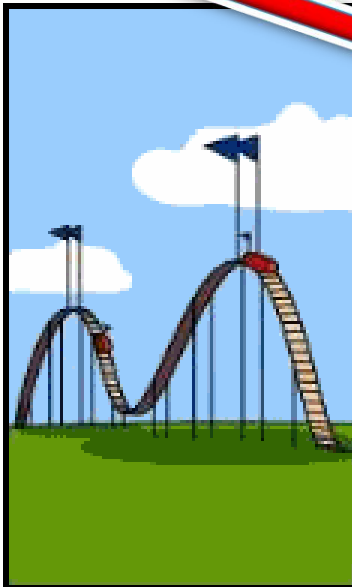
How the customer was billed
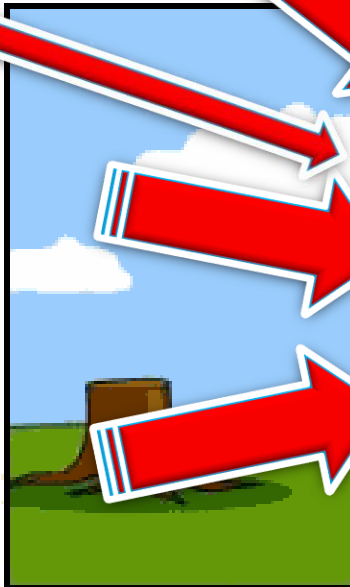
How it was supported
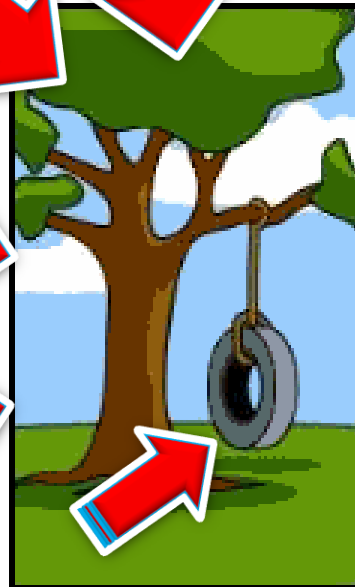
What the customer really needed
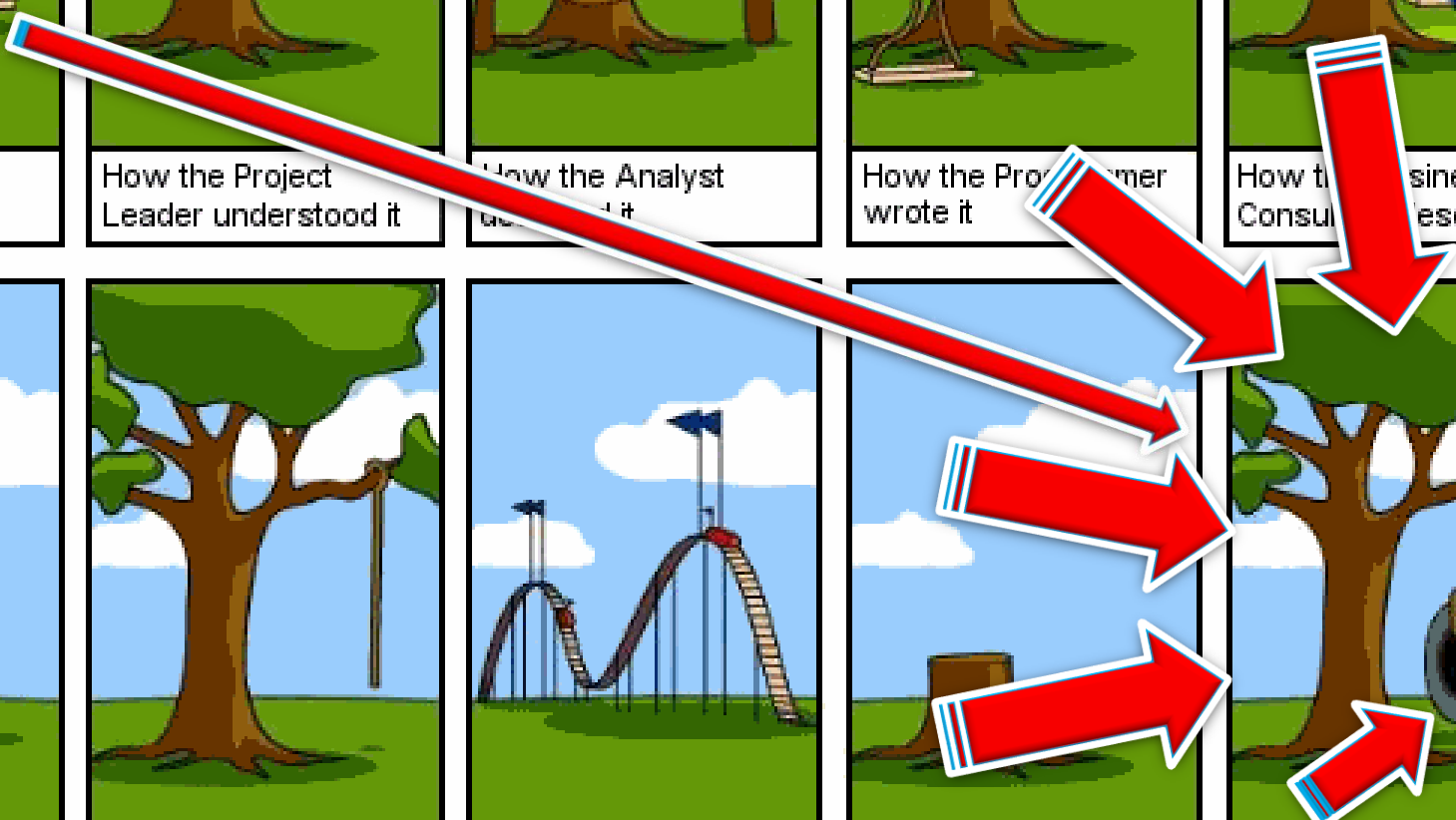
Definition of Requirements Engineering

Not a phase or stage!

Communication is as important as the analysis

Quality means fitness-for-purpose. Cannot say anything about quality unless you understand the purpose

Requirements Engineering (RE) is a set of activities concerned with identifying and communicating the purpose of a software-intensive system, and the contexts in which it will be used. Hence, RE acts as the bridge between the real world needs of users, customers, and other constituencies affected by a software system, and the capabilities and opportunities afforded by software-intensive technologies

Designers need to know how and where the system will be used

Requirements are partly about what is needed...

...and partly about what is possible

Need to identify all the stakeholders - not just the customer and user

What is a Requirement

- It may range from a high-level abstract statement of a service or of a system constraint to a detailed mathematical functional specification.
- This is inevitable as requirements may serve a dual function
  - May be the basis for a bid for a contract - therefore must be open to interpretation;
  - May be the basis for the contract itself - therefore must be defined in detail;
  - Both these statements may be called requirements.

Requirement Abstraction

- "If a company wishes to let a contract for a large software development project, it must define its needs in a sufficiently abstract way that a solution is not pre-defined. The requirements must be written so that several contractors can bid for the contract, offering, perhaps, different ways of meeting the client organization's needs. Once a contract has been awarded, the contractor must write a system definition for the client in more detail so that the client understands and can validate what the software will do. Both of these documents may be called the requirements document for the system."

Case Study

- The Mentcare system shall generate monthly management reports showing the cost of drugs prescribe by each clinic during that month.
- On the last working day of each month, a summary of the drugs prescribed, their cost and the prescribing clinics shall be generated.
- The system shall generate the report for printing after 17.30 on the last working day of the month.
- A report shall be created for each clinic and shall list the individual drug names, the total number of prescriptions, the number of doses prescribed and the total cost of the prescribed rugs.
- If drugs are available in different dose units (e.g. 10mg, 20mg, etc.) separate reports shall be created for each dose unit.
- Access to drug cost reports shall be restricted to authorized users as listed on a management access control list.

- The Mentcare will have patients whose information will be recorded in the system.
- It will have doctors who will be responsible for assessing and treating patients.
- Nurses who will coordinate the consultations with doctors and administer some treatments.
- Medical receptionists who will manage patients' appointments.
- IT staff who will responsible for installing and maintaining the system.
- A medical ethics manager who must ensure that the system meets current ethical guidelines for patient care.
- Health care managers who obtain management information from the system.
- Medical records staff who are responsible for ensuring that system information can be maintained and preserved, and that record keeping procedures have been properly implemented.

- A user shall be able to search the appointments lists for all clinics.
- The system shall generate each day, for each clinic, a list of patients who are expected to attend appointments that day.
- Each staff member using the system shall be uniquely identified by his or her 8-digit employee number.
- The Mentcare system shall be available to all clinics during normal working hours (Mon–Fri, 0830–17.30). Downtime within normal working hours shall not exceed five seconds in any one day.
- Users of the Mentcare system shall authenticate themselves using their health authority identity card.
- The system shall implement patient privacy provisions as set out in HStan-03-2006-priv.

Types of Requirements

- User requirements:
  - Written for Customer

Types of Requirements

- System requirements:
  - Functional and Non-functional Requirements
  - May be part of contract between client and contractor

System Function's

Services and Operational Constraints

System stakeholders

- Any person or organization who is affected by the system in some way and so who has a legitimate interest
- Stakeholder types
  - End users
  - System managers
  - System owners
  - External stakeholders

Readers of Different Types of Requirement Specification

User Requirements	→	Client managers System end-users Client engineers Contractor managers System Architect

System Requirements	→	System end -users Client engineers System Architect Software Developers

Functional Requirements

Non-Functional Requirements
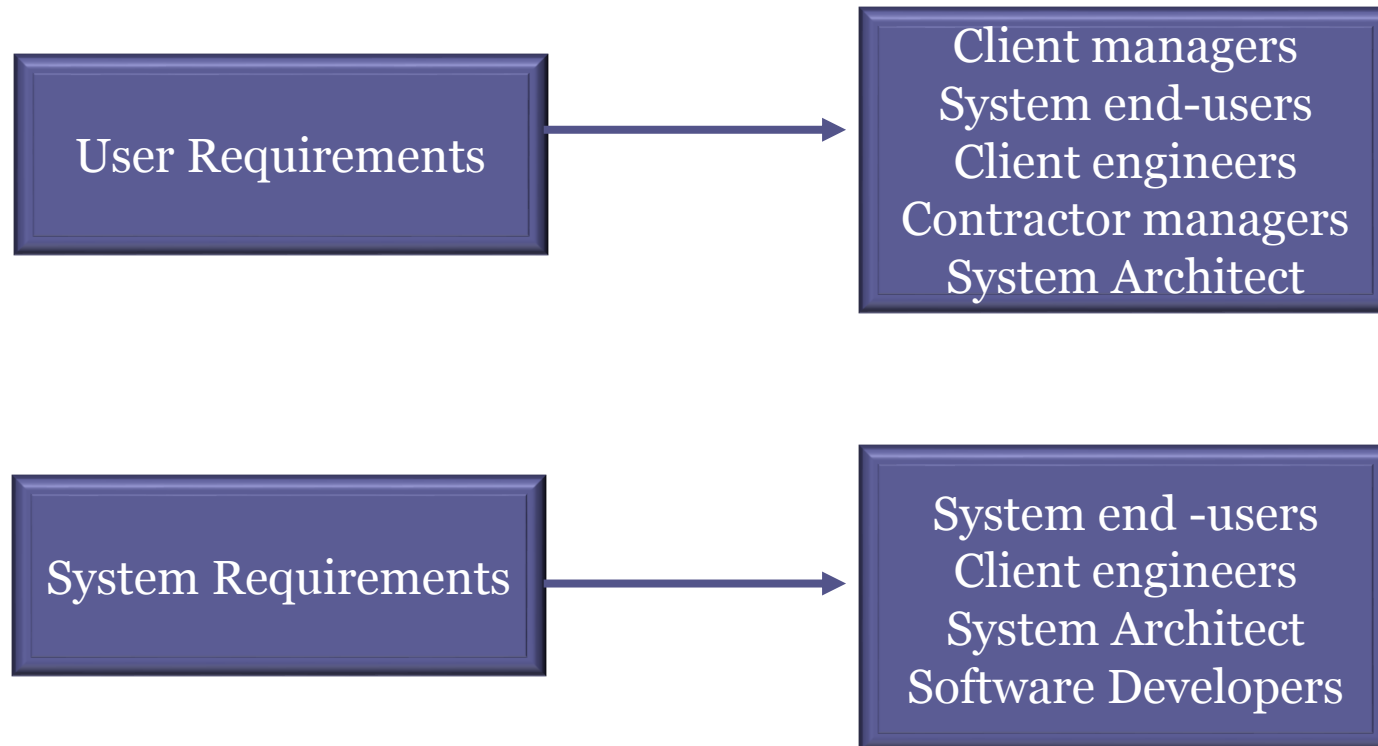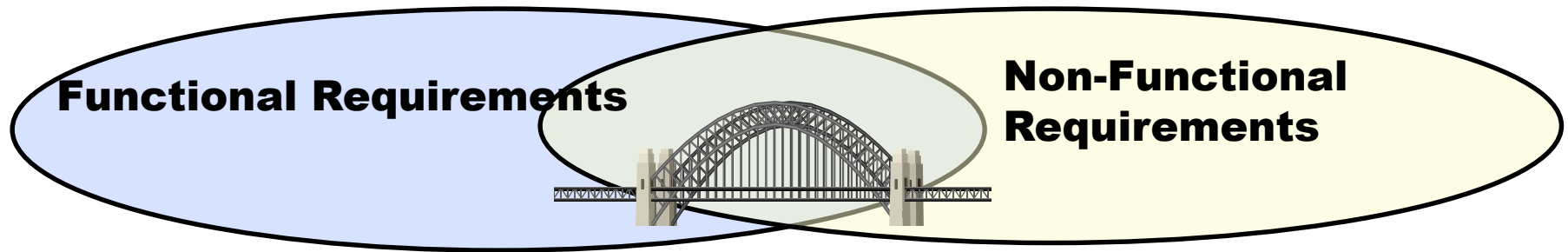
- Functional Requirements:
  - How system should react to particular inputs .
  - How system should behave in particular situations.
- Non-functional Requirements:
  - Constraints on the services
  - Functions offered by the system

Non-functional
requirements

Product requirements — Organizational requirements — External requirements

Efficiency requirements, Dependability requirements, Security requirements, Regulatory requirements, Ethical requirements

Usability requirements

Environmental requirements, Operational requirements, Development requirements, Legislative requirements

Performance requirements, Space requirements

Accounting requirements, Safety/security requirements

Requirements engineering processes

- The processes used for RE vary widely depending on the application domain, the people involved and the organization developing the requirements.
- However, there are a number of generic activities common to all processes
  - Requirements elicitation;
  - Requirements analysis;
  - Requirements validation;
  - Requirements management.
- In practice, RE is an iterative activity in which these processes are interleaved.

A spiral view of the requirements engineering process
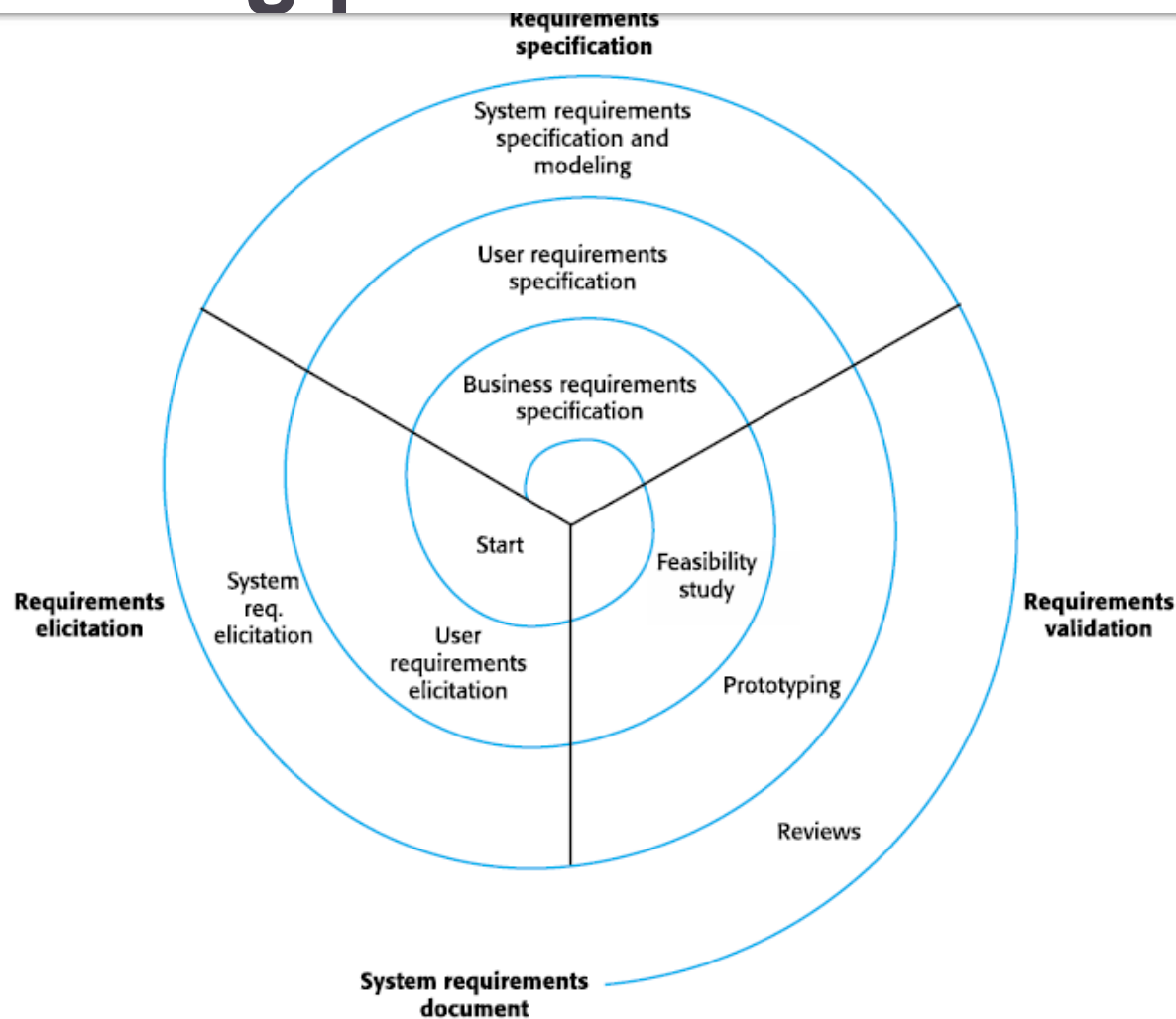
Requirements elicitation and analysis

- Sometimes called requirements elicitation or requirements discovery.
- Involves technical staff working with customers to find out about the application domain, the services that the constraints.
- May involve end-users, managers, engineers involved in maintenance, domain experts, trade unions, etc. These are called *stakeholders*.

Requirement Elicitation

- Software engineers work with a range of system stakeholders to find out about the application domain, the services that the system should provide, the required system performance, hardware constraints, other systems, etc.
- Stages include:
  - Requirements discovery,
  - Requirements classification and organization,
  - Requirements prioritization and negotiation,
  - Requirements specification.

Problem with Requirements Elicitation

- Stakeholders don't know what they really want.
- Stakeholders express requirements in their own terms.
- Different stakeholders may have conflicting requirements.
- Organizational and political factors may influence the system requirements.
- The requirements change during the analysis process. New stakeholders may emerge and the business environment may change.

The requirement Elicitation and Analysis Process
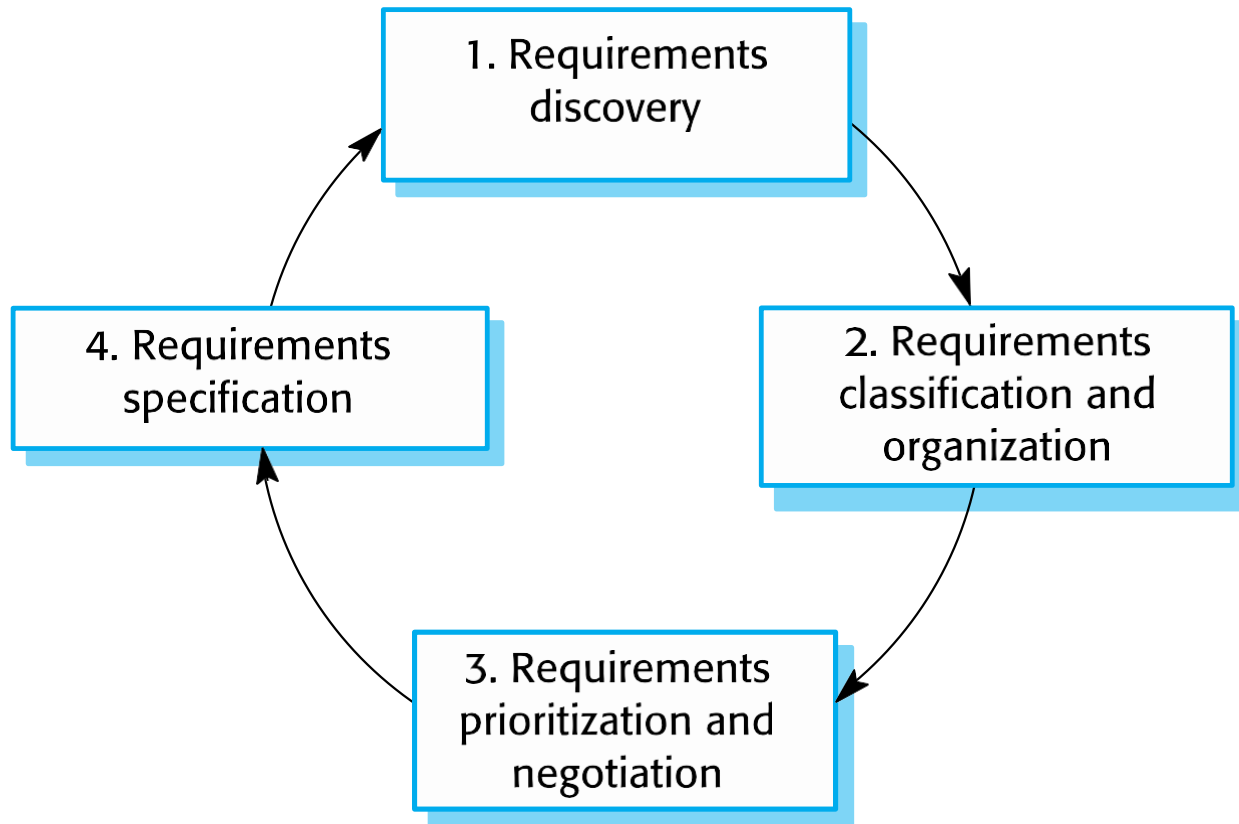
Interviews

- Types:
  - Structured - agenda of fairly open questions
  - Open-ended - no pre-set agenda
- Advantages
  - Rich collection of information
    - Good for uncovering opinions, feelings, goals, as well as hard facts
  - Can probe in depth, & adapt follow up questions to what the person tells you
- Disadvantages
  - Large amount of qualitative data can be hard to analyze
  - Hard to compare different respondents
  - Interviewing is a difficult skill to master
- Watch for
  - Unanswerable questions ("how do you tie your shoelaces?")
  - Tacit knowledge (and post-hoc rationalizations)
  - Removal from context
  - Interviewer's attitude may cause bias (e.g. variable attentiveness)

Ethnography

- A social scientist spends a considerable time observing and analyzing how people actually work.

- People do not have to explain or articulate their work.

- Social and organizational factors of importance may be observed.

- Ethnographic studies have shown that work is usually richer and more complex than suggested by simple system models.

- Ethnography is effective for understanding existing processes but cannot identify new features that should be added to a system.

Ethnography and prototyping for requirements analysis
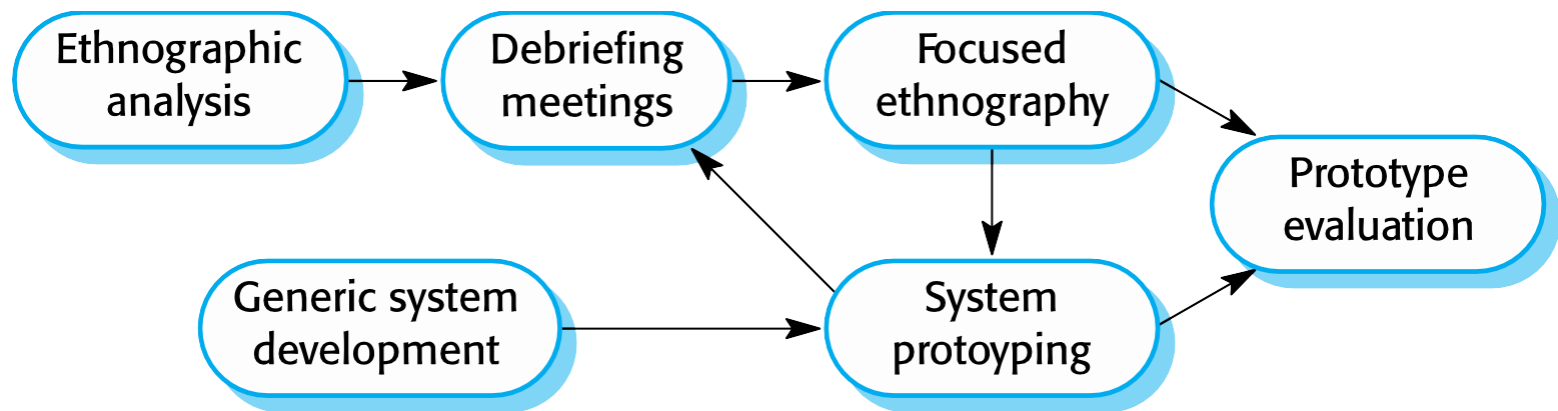
Stories and scenarios

- Scenarios and user stories are real-life examples of how a system can be used.

- Stories and scenarios are a description of how a system may be used for a particular task.

- Because they are based on a practical situation, stakeholders can relate to them and can comment on their situation with respect to the story.

Scenarios

- A structured form of user story

- Scenarios should include
  - A description of the starting situation;
  - A description of the normal flow of events;
  - A description of what can go wrong;
  - Information about other concurrent activities;
  - A description of the state when the scenario finishes.

Requirements specification

Requirements specification

- The process of writing down the user and system requirements in a requirements document.

- User requirements have to be understandable by end-users and customers who do not have a technical background.

- System requirements are more detailed requirements and may include more technical information.

- The requirements may be part of a contract for the system development
  - It is therefore important that these are as complete as possible.

Ways of writing a system requirements specification

Notation	Description
Natural language	The requirements are written using numbered sentences in natural language. Each sentence should express one requirement.
Structured natural language	The requirements are written in natural language on a standard form or template. Each field provides information about an aspect of the requirement.
Design description languages	This approach uses a language like a programming language, but with more abstract features to specify the requirements by defining an operational model of the system. This approach is now rarely used although it can be useful for interface specifications.
Graphical notations	Graphical models, supplemented by text annotations, are used to define the functional requirements for the system; UML use case and sequence diagrams are commonly used.
Mathematical specifications	These notations are based on mathematical concepts such as finite-state machines or sets. Although these unambiguous specifications can reduce the ambiguity in a requirements document, most customers don't understand a formal specification. They cannot check that it represents what they want and are reluctant to accept it as a system contract

Requirements and design

- In principle, requirements should state what the system should do and the design should describe how it does this.

- In practice, requirements and design are inseparable
  - A system architecture may be designed to structure the requirements;
  - The system may inter-operate with other systems that generate design requirements;
  - The use of a specific architecture to satisfy non-functional requirements may be a domain requirement.
  - This may be the consequence of a regulatory requirement.

Use cases

- Use-cases are a kind of scenario that are included in the UML.

- Use cases identify the actors in an interaction and with h describe the interaction itself.

- A set of use cases should describe all possible interactions with the system.

- High-level graphical model supplemented by more detailed tabular description.

- UML sequence diagrams may be used to add detail to use-cases by showing the sequence of event processing in the system.