COMP41670 Software Engineering

13. Requirements Engineering

Dr Avishek Nag

UCD School of Computer Science. **Scoil na Ríomheolaíochta UCD.**

Table of Contents

What is Requirements Engineering?

Requirements Engineering

- .. is the process of establishing the services that a customer requires from a system and the constraints under which it operates and is developed.
- A requirement may range from a high-level abstract statement of a service or of a system constraint to a detailed mathematical functional specification.

Business Aspects

- The high-level requirements specification may be used as the basis of a tendering processes whereby the customer seeks tenders from software development companies for building the system.
- Software development companies submit tenders (bids) for the implementation project. The bid generally includes a price, an outline implementation plan and a profile of the company's previous experience in the area.
- The customer reviews the tenders and selects the provider.
- The requirements specification becomes part of the contract between the customer and the software developer.

Requirements Specification

- … documents the user and system requirements.
- Has to be understandable by end-users and customers.
- May be part of the contract.
- Should state what the system will do, not how it will do it.

Types of Requirements

- **User requirements**: Statements in natural language plus diagrams of the services the system provides and its operational constraints. Written for customers.
- **System requirements**: A structured document setting out detailed descriptions of the system's functions, services and operational constraints. Defines what should be implemented so may be part of a contract between client and contractor.

Mentcare Example: Requirements

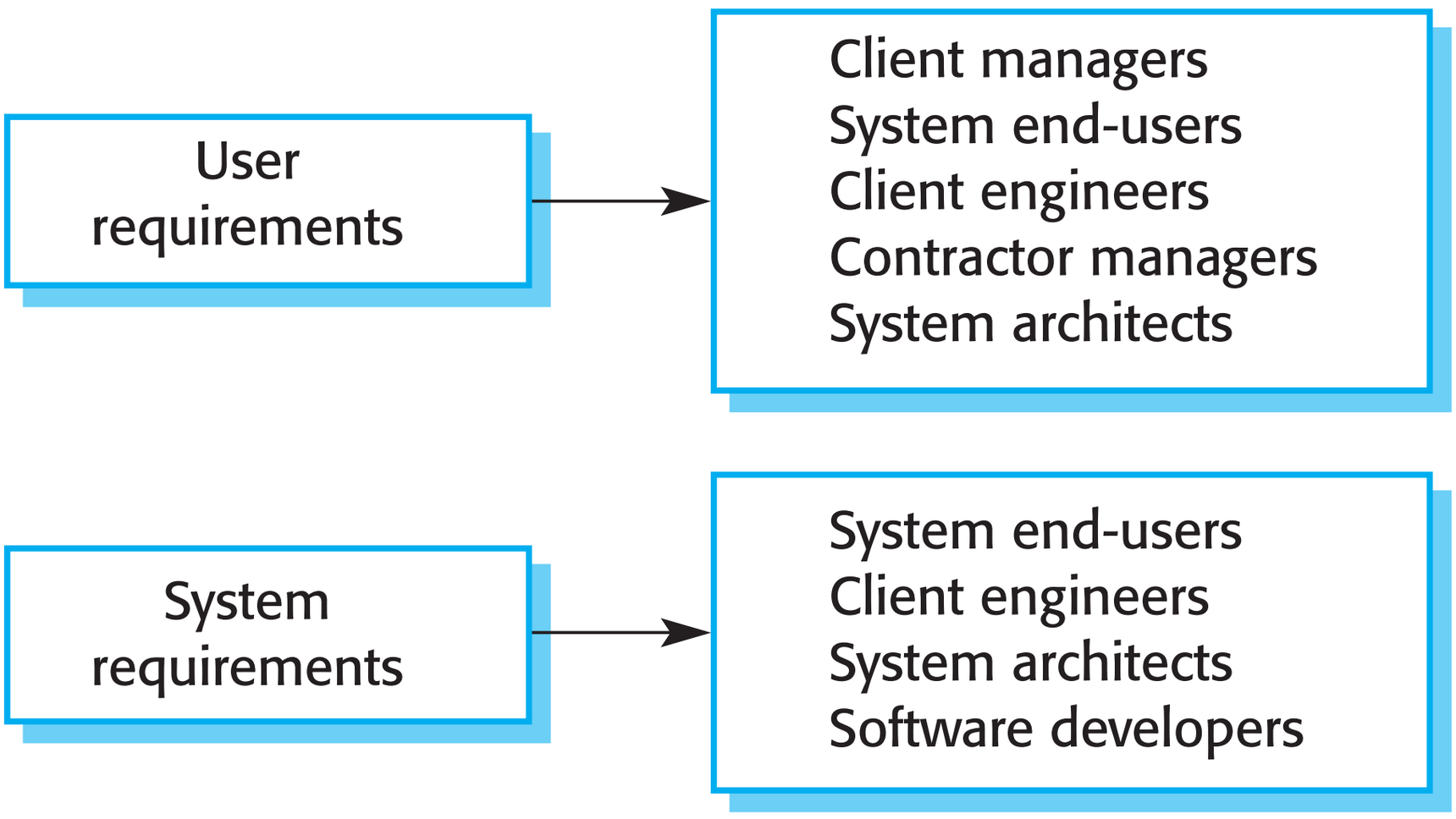
User requirements definition

1. The Mentcare system shall generate monthly management reports showing the cost of drugs prescribed by each clinic during that month.

System requirements specification

1.1 On the last working day of each month, a summary of the drugs prescribed, their cost and the prescribing clinics shall be generated.
1.2 The system shall generate the report for printing after 17.30 on the last working day of the month.
1.3 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 drugs.
1.4 If drugs are available in different dose units (e.g. 10mg, 20mg, etc) separate reports shall be created for each dose unit.
1.5 Access to drug cost reports shall be restricted to authorized users as listed on a management access control list.

Requirements Readership



Mentcare Example: Stakeholders

- *Patients whose information is recorded in the system.*
- *Doctors who are responsible for assessing and treating patients.*
- *Nurses who coordinate the consultations with doctors and administer some treatments.*
- *Medical receptionists who manage patients' appointments.*
- *IT staff who are 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.*

Functional and Non-Functional 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.
- Essentially, states what the system should do.
- May state what the system should not do.
- **Non-functional requirements**: Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.
- Often apply to the system as a whole rather than individual features or services.
- **Domain requirements**: Constraints on the system from the domain of operation

Mentacare Example

- *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.*

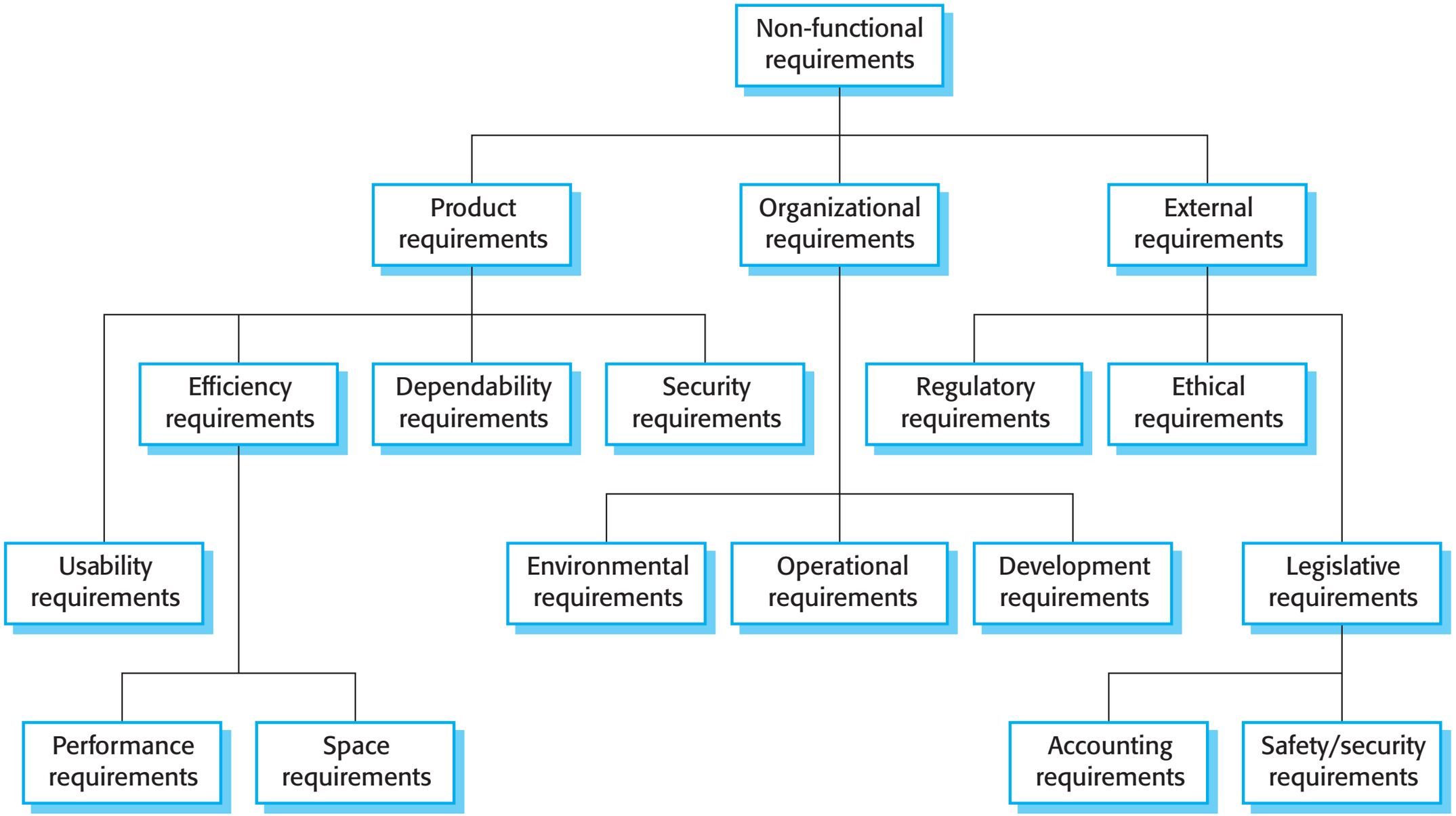
Requirements Quality

- Requirements should be complete, consistent and unambiguous.
- Anything missing or misunderstood causes big problems downstream (i.e. later in the project).
- Unfortunately, complete, consistent and unambiguous is almost impossible… :-(

Non-Functional Requirements

- These define system properties and constraints e.g. reliability, response time and storage requirements. Constraints are I/O device capability, system representations, etc.
- Process requirements may also be specified mandating a particular IDE, programming language or development method.
- Non-functional requirements may be more critical than functional requirements. If these are not met, the system may be useless.

Non-Functional Requirements



Mentcare Example: Non-Functional Requirements

- *Product requirement: 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.*

- *Organizational requirement: Users of the Mentcare system shall authenticate themselves using their health authority identity card.*

- *External requirement: The system shall implement patient privacy provisions as set out in HStan-03-2006-priv.*

Goals and Requirements

- **Goal**: A general intention to achieve something.
- **Verifiable**: A measurable objective.

Mentcare Example: Goals and Requirements

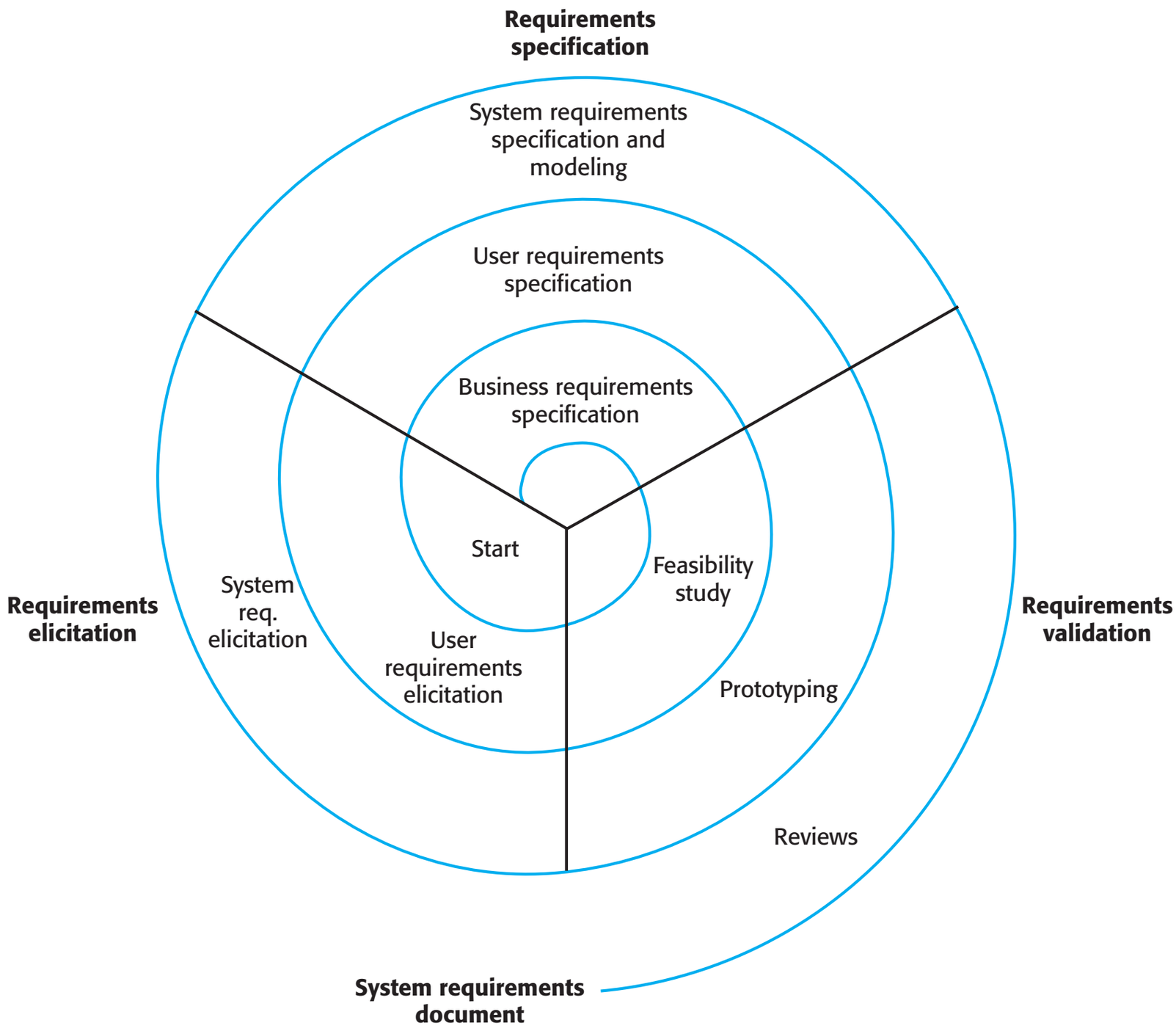
- *The system should be easy to use by medical staff and should be organized in such a way that user errors are minimised. (Goal)*
- *Medical staff shall be able to use all the system functions after four hours of training. After this training, the average number of errors made by experienced users shall not exceed two per hour of system use. (Testable non-functional requirement)*

Requirements Engineering Process

Requirements Engineering Processes

- In general, the activities are:
  - **Requirements elicitation**: technical staff work with customers and stakeholders to find out about the application.
  - **Requirements analysis**: Organise, prioritise and document the requirements.
  - **Requirements validation**: Check that the written requirements are what the customer really wants.
  - **Requirements management:** Dealing with changes in the requirements. Re-work will impact on cost and schedule.

Spiral Model



Requirements Elicitation: Problems

- Stakeholders don’t know what they really want.
- Stakeholders use their own terminology.
- Stakeholders have conflicting requirements.
- Organisational and political factors influence matters.
- The business environment may change, leading to changed requirements.

Requirements Elicitation

- **User stories** are an approach to Requirements Elicitation.
- Based on asking users:
  - What is their role with respect to the software?
  - What they want the software to do for them?
  - What is the benefit to them?
- Don’t talk about how the software will work.

User Stories: Understanding What the User Wants

- Three C's:
  - Conversation: Between development team, product owner, users, and stakeholders. To understand what the application should do. Workshops are often used for this.
  - Cards: Document the requirements as **user stories**. A user story specifies was class of users want to achieve (goal) and why they want to achieve it (benefit).
  - Confirmation: Document the acceptance criteria that clarify the desired behaviour.

- Initial user stories are refined to clarify and add detail to what the application should do.

User Stories: Understanding What the User Wants

Template

<User Story Title>

As a <user role> I want to <goal> so that <benefit>.

Conditions of Satisfaction

Verify that ...
Verify that ....

Examples

Find Reviews Near Me

As a typical user I want to see unbiased reviews of restaurants near my location so that I can pick a good restaurant for dinner.

Conditions of Satisfaction

Verify that all restaurants are listed in order of distance from me.
Verify that I don't need to enter my current position.

User Stories: Understanding What the User Wants

- User Stories should be INVEST:
  - Independent of each other.
  - Negotiable. Users should say *what* they want, the team should figure out *how* to implement it.
  - Valuable to the customer or user.
  - Estimatable. The team should be able to estimate the cost implementing the user story.
  - Small. Stories should be broken down so that they can be built by the team within a sprint (1-2 weeks), i.e. they are sprintable.
  - Testable. The acceptance criteria (derived from the conditions of satisfaction) from the should be clear.

Requirements Analysis

- Merge repeated user stories.
- Group user stories.
- Prioritise user stories.

Story Map

Epics
(large stories)

Themes
(medium stories)

Stories
(small)

Release roadmap

Requirements Validation

- Checks that:
  - The system best supports the user’s needs
  - There are no requirements conflicts
  - Everything is included
  - The requirements can be implemented given the available budget
  - The team can check that the requirements have been met

- Checks by means of:
  - Document reviews by / with the stakeholders
  - Showing prototypes to the stakeholders

Requirements Management.

- … is the process of managing changing requirements during the requirements engineering process and system development.
- New requirements emerge as a system is being developed and after it has gone into use.
- Need to keep track of requirement changes using a change management process.
  - Change proposal written
  - Change proposal submitted and logged
  - Change analysis: proposal costed and impact on schedule assessed
  - Change review: stakeholder committee makes yes/no decision

Pop Quiz

Pop Quiz

- In your group, do a client interview to create a user story.
- The client wants a system for a UCD Club or Soc. Could be an app for members or for leaders.
  - Pen and paper
  - http://www.umletino.com
  - https://lucid.app
- Keep what you have for next time

How To Conduct User Interviews Like A Pro (UX Design), CareerFoundry, YouTube

Prototyping

Rapid Prototyping: Sketching, Google for Startups, YouTube

Rapid Prototyping: Digital, Google for Startups, YouTube

Rapid Prototyping: Digital, Google for Startups, YouTube

Summary

- Requirements for a software system set out what the system should do and define constraints on its operation and implementation.
- Functional requirements are statements of the services that the system must provide or are descriptions of how some computations must be carried out. Non-functional requirements often constrain the system being developed and the development process being used.
- Requirements elicitation identifies what users want the system to do.
- Requirements specification formally documents the user and system requirements.
- Requirements validation is checking that the requirements specification is correct.