



澳門理工學院
Instituto Politécnico de Macau
Macao Polytechnic Institute

COMP223: Software Engineering Requirement Engineering

Dr. Kim, Song-Kyoo (Amang)
Associate Professor,

Computer Science Program
MACAO POLYTECHNIC INSTITUTE
Macau, SAR





澳門理工學院
Instituto Politécnico de Macau
Macao Polytechnic Institute

Requirement Engineering (Part 1)



Session Objectives



- Functional and non-functional requirements.
- Requirements engineering processes.
- Requirements elicitation.
- Requirements specification.
- Requirements validation.
- Requirements change.

Introduction (1/9)



- Requirement Engineering

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

Introduction (2/9)



● 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.

Introduction (3/9)



- Requirements abstraction (Davis)

“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.”

Introduction (4/9)



- Types of requirement

- **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.

Introduction (5/9)



● User and system 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.

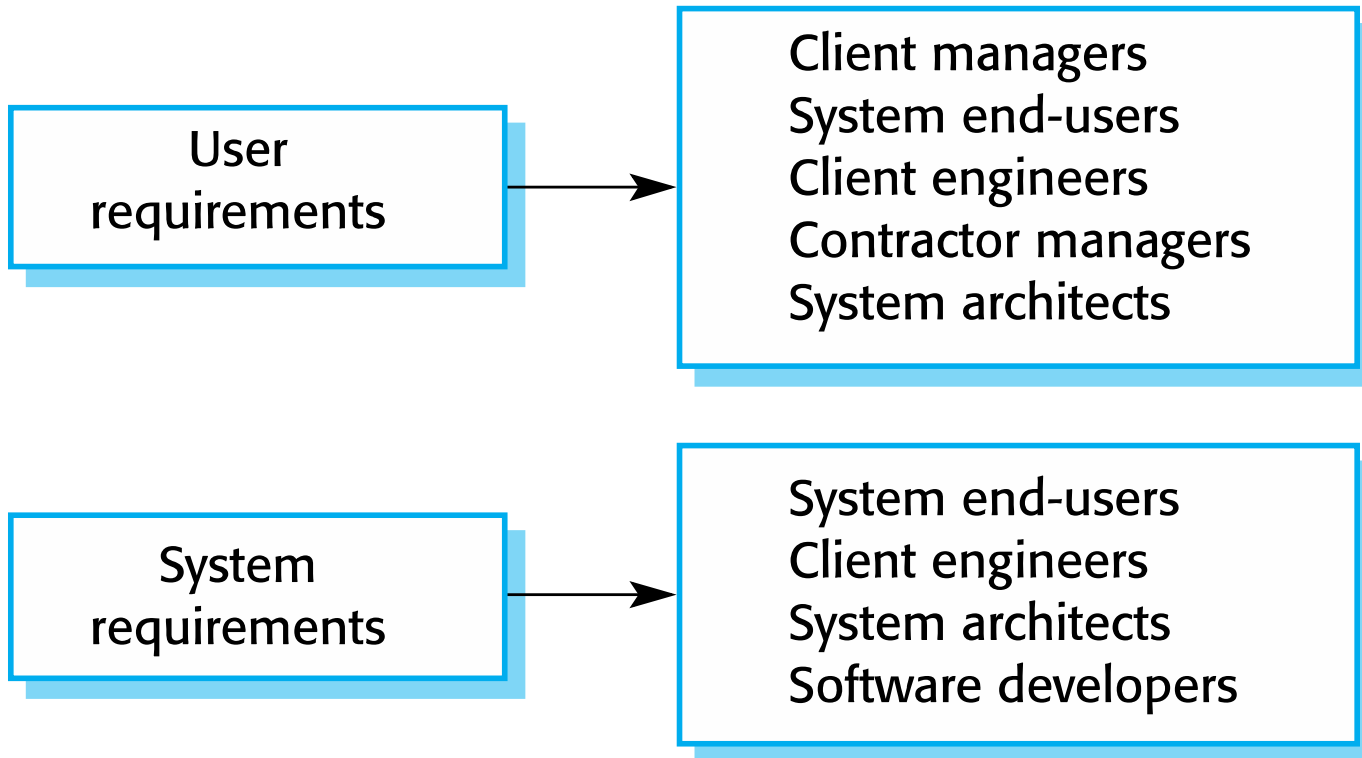
System requirements specification

- 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.

Introduction (6/9)



- Readers of different types of requirements specification



Introduction (7/9)



- 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

Introduction (8/9)



- Stakeholders in the Mentcare (Healthcare) system:
 - 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.

Introduction (9/9)



- Agile methods and requirements:
 - Many agile methods argue that producing detailed system requirements is a waste of time as requirements change so quickly.
 - The requirements document is therefore always out of date.
 - Agile methods usually use incremental requirements engineering and may express requirements as user stories.
 - This is practical for business systems but problematic for systems that require pre-delivery analysis (e.g. critical systems) or systems developed by several teams.

Functional & Non-Functional Req. (1/12)



● 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.
- 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.

Functional & Non-Functional Req. (2/12)



● Functional requirements

- Describe functionality or system services.
- Depend on the type of software, expected users and the type of system where the software is used.
- Functional user requirements may be high-level statements of what the system should do.
- Functional system requirements should describe the system services in detail.
- **(eg.) Mentcare system:**
 - 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.

Functional & Non-Functional Req. (3/12)



● Requirements imprecision:

- Problems arise when functional requirements are not precisely stated.
- Ambiguous requirements may be interpreted in different ways by developers and users.
- Consider the term 'search' in requirement 1
 - User intention – search for a patient name across all appointments in all clinics;
 - Developer interpretation – search for a patient name in an individual clinic. User chooses clinic then search.

Functional & Non-Functional Req. (4/12)



- Requirements completeness and consistency:
 - In principle, requirements should be both complete and consistent.
 - **Complete**: They should include descriptions of all facilities required.
 - **Consistent**: There should be no conflicts or contradictions in the descriptions of the system facilities.
 - In practice, because of system and environmental complexity, it is impossible to produce a complete and consistent requirements document.

Functional & Non-Functional Req. (5/12)



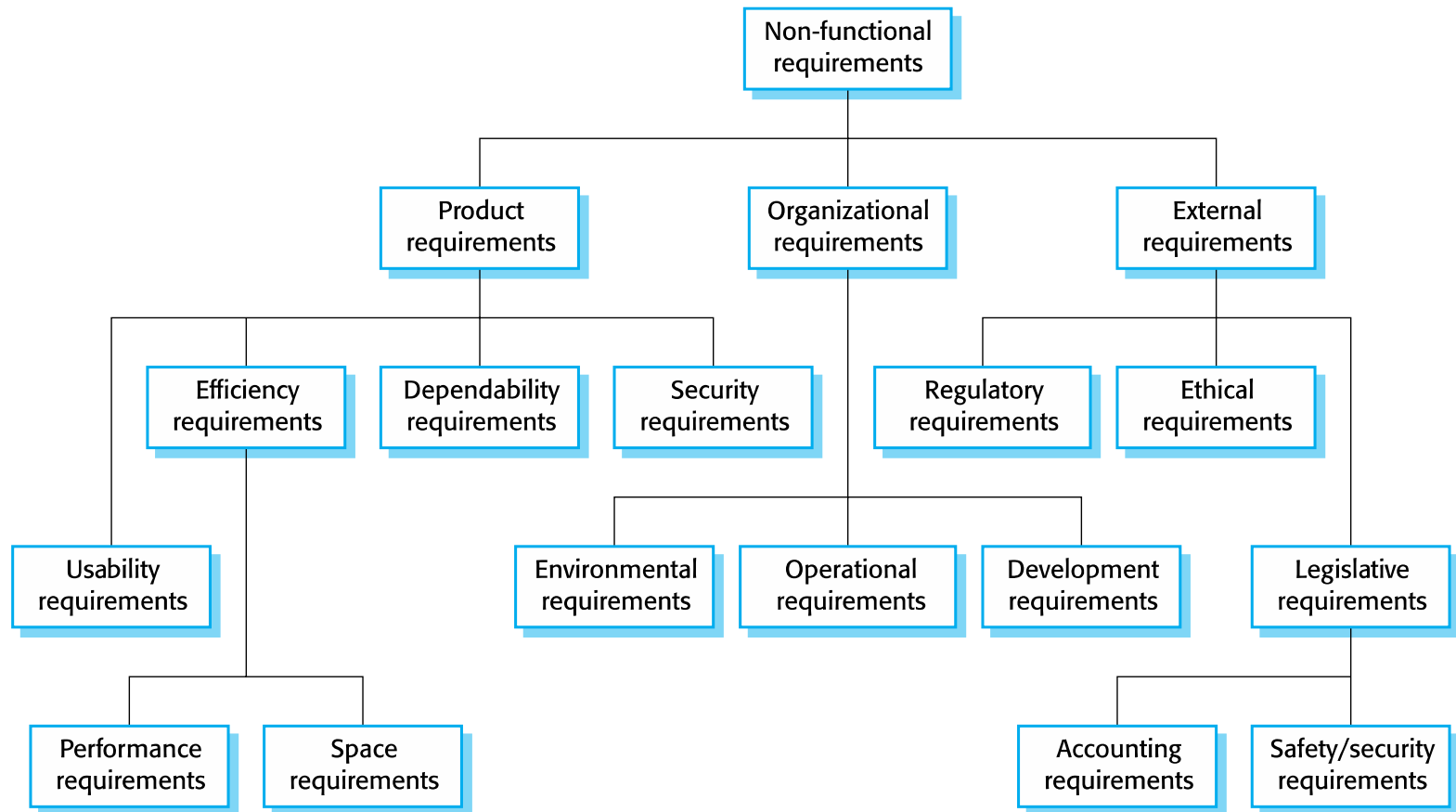
● Non-functional requirements

- 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.

Functional & Non-Functional Req. (6/12)



● Types of nonfunctional requirement



Functional & Non-Functional Req. (7/12)



- Non-functional requirements implementation
 - Non-functional requirements may affect the overall architecture of a system rather than the individual components.
 - For example, to ensure that performance requirements are met, you may have to organize the system to minimize communications between components.
 - A single non-functional requirement, such as a security requirement, may generate a number of related functional requirements that define system services that are required.
 - It may also generate requirements that restrict existing requirements.

Functional & Non-Functional Req. (8/12)



- Non-functional classifications

- **Product requirements:** Requirements which specify that the delivered product must behave in a particular way e.g. execution speed, reliability, etc.
- **Organizational requirements:** Requirements which are a consequence of organizational policies and procedures e.g. process standards used, implementation requirements, etc.
- **External requirements:** Requirements which arise from factors which are external to the system and its development process e.g. interoperability requirements, legislative requirements, etc.

Functional & Non-Functional Req. (9/12)



- Examples of nonfunctional requirements in the Mentcare system

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.

Functional & Non-Functional Req. (10/12)



- Goals and requirements

- Non-functional requirements may be very difficult to state precisely and imprecise requirements may be difficult to verify.
- **Goal** -- A general intention of the user such as ease of use.
- **Verifiable non-functional requirement** -- A statement using some measure that can be objectively tested.
- Goals are helpful to developers as they convey the intentions of the system users.

Functional & Non-Functional Req. (11/12)



● Usability requirements

- The system should be easy to use by medical staff and should be organized in such a way that user errors are minimized. (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)

Functional & Non-Functional Req. (12/12)



● Metrics for specifying nonfunctional requirements

Property	Measure
Speed	Processed transactions/second User/event response time Screen refresh time
Size	Mbytes Number of ROM chips
Ease of use	Training time Number of help frames
Reliability	Mean time to failure Probability of unavailability Rate of failure occurrence Availability
Robustness	Time to restart after failure Percentage of events causing failure Probability of data corruption on failure
Portability	Percentage of target dependent statements Number of target systems

Requirements engineering processes (1/2)

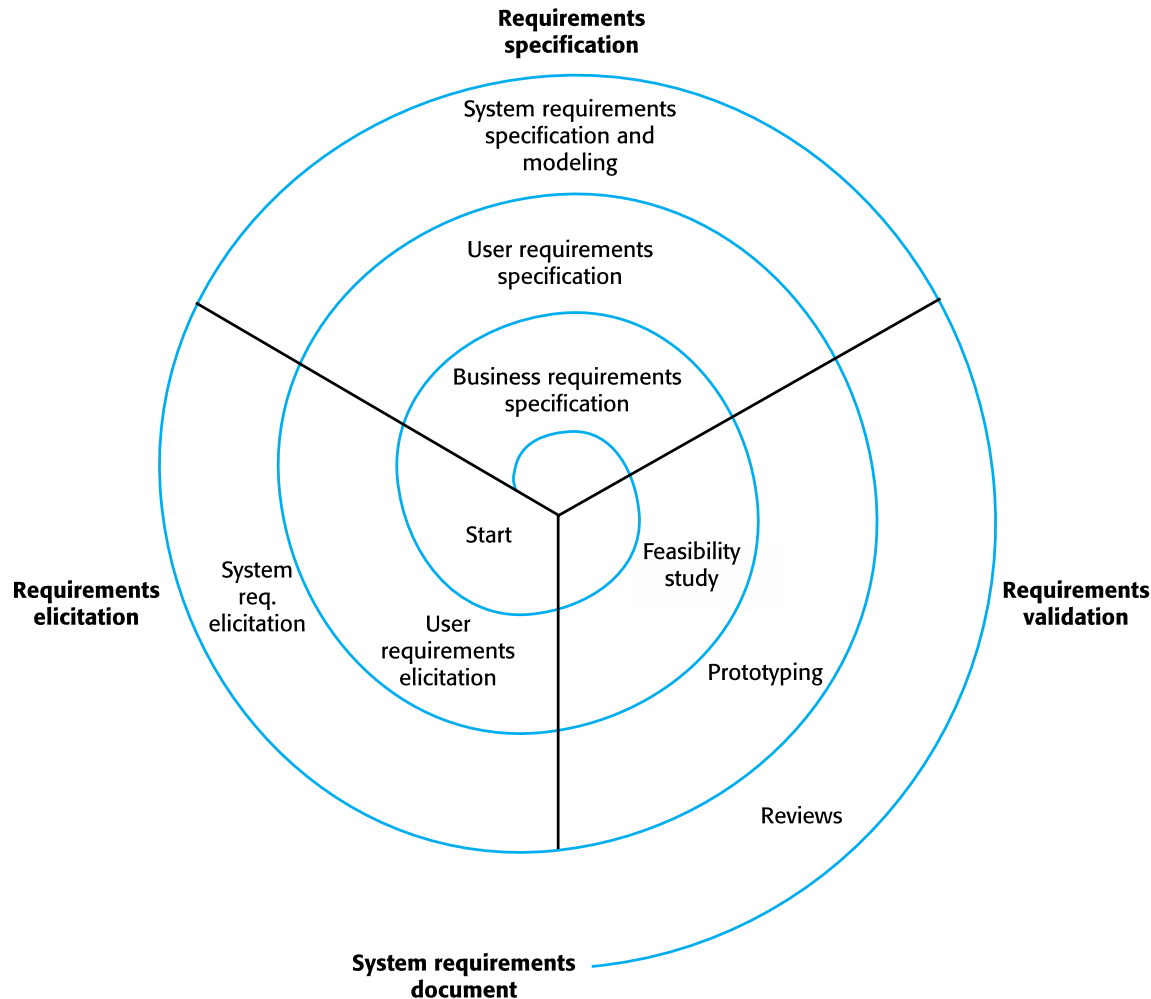


- The processes used for RE vary widely depending on the application domain, the people involved and the organization developing the requirements.
- However, there are a number of generic activities common to all processes
 - Requirements elicitation (define);
 - Requirements analysis;
 - Requirements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

Requirements engineering processes (2/2)



- A spiral view of the requirements engineering process



Requirements elicitation (1/18)



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

Requirements elicitation (2/18)



- 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.

Requirements elicitation (3/18)



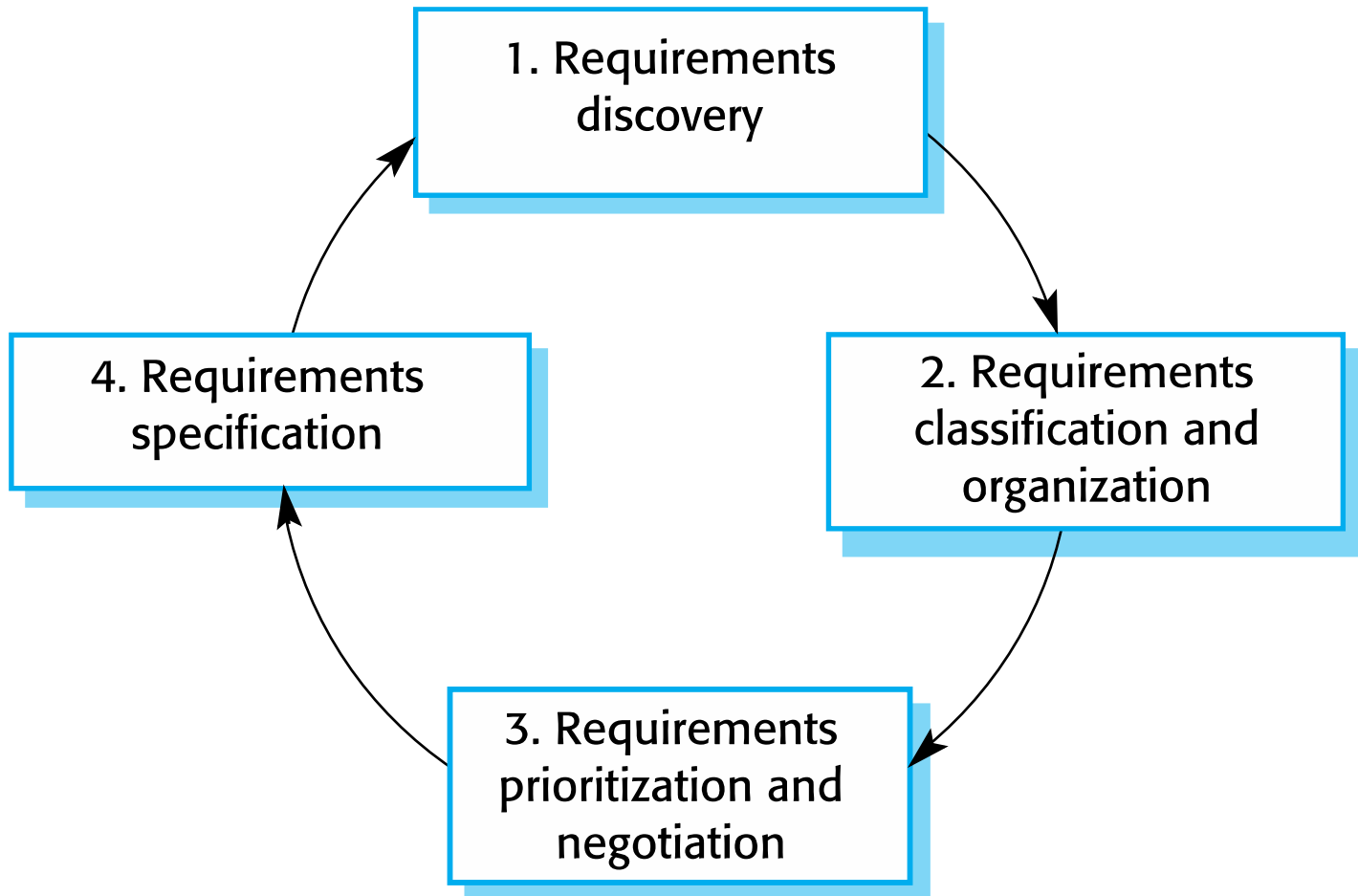
● Problems of requirements elicitation

- Stakeholders do not 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.

Requirements elicitation (4/18)



- The requirements elicitation and analysis process



Requirements elicitation (5/18)



● Process activities:

- **Requirements discovery**: Interacting with stakeholders to discover their requirements. Domain requirements are also discovered at this stage.
- **Requirements classification and organization**: Groups related requirements and organizes them into coherent clusters.
- **Prioritization and negotiation**: Prioritizing requirements and resolving requirements conflicts.
- **Requirements specification**: Requirements are documented and input into the next round of the spiral.

Requirements elicitation (6/18)



- Requirements discovery
 - The process of gathering information about the required and existing systems and distilling the user and system requirements from this information.
 - Interaction is with system stakeholders from managers to external regulators.
 - Systems normally have a range of stakeholders.
- Request For Information (RFI)
 - Typical way to discover the requirements from customers and/or suppliers (manufacturers).
 - Good substitute of other requirement discovery methods.

Requirements elicitation (7/18)



- Interviewing

- Formal or informal interviews with stakeholders are part of most RE processes.

- Types of interview

- Closed interviews based on pre-determined list of questions.
- Open interviews where various issues are explored with stakeholders.

- Effective interviewing

- Be open-minded, avoid pre-conceived ideas about the requirements and are willing to listen to stakeholders.
- Prompt the interviewee to get discussions going using a springboard question, a requirements proposal, or by working together on a prototype system.

Requirements elicitation (8/18)



● Interviews in practice

- Normally a mix of closed and open-ended interviewing.
- Interviews are good for getting an overall understanding of what stakeholders do and how they might interact with the system.
- Interviewers need to be open-minded without pre-conceived ideas of what the system should do
- You need to prompt the use to talk about the system by suggesting requirements rather than simply asking them what they want.

Requirements elicitation (9/18)



● Problems with interviews

- Application specialists may use language to describe their work that is not easy for the requirements engineer to understand.
- Interviews are not good for understanding domain requirements
- Requirements engineers cannot understand specific domain terminology;
- Some domain knowledge is so familiar that people find it hard to articulate or think that it isn't worth articulating.

Requirements elicitation (10/18)



● Ethnography (social ecology)

- 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.

Requirements elicitation (11/18)



● Scope of ethnography

- Requirements that are derived from the way that people actually work rather than the way which process definitions suggest that they ought to work.
- Requirements that are derived from cooperation and awareness of other activities.
- Awareness of what other people are doing leads to changes in the ways in which we do things.

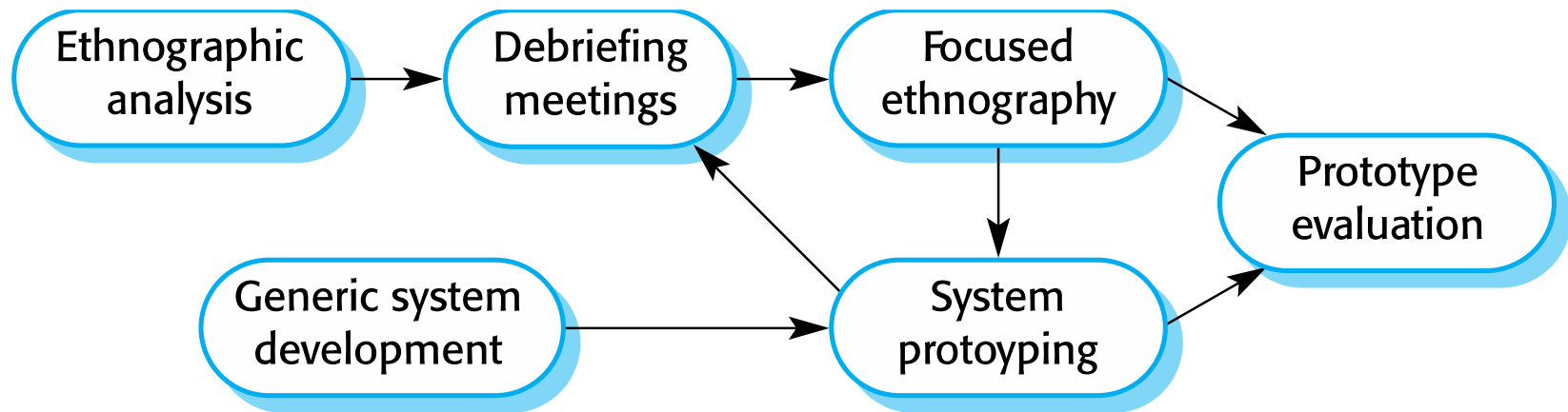
● Focused ethnography

- Developed in a project studying the air traffic control process
- Combines ethnography with prototyping.
- Prototype development results in unanswered questions which focus the ethnographic analysis.

Requirements elicitation (12/18)



- Ethnography and prototyping for requirements analysis



Requirements elicitation (13/18)



● 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.

Requirements elicitation (14/18)



● Photo sharing in the classroom (iLearn)

Jack is a primary school teacher in Ullapool (a village in northern Scotland). He has decided that a class project should be focused around the fishing industry in the area, looking at the history, development and economic impact of fishing. As part of this, pupils are asked to gather and share reminiscences from relatives, use newspaper archives and collect old photographs related to fishing and fishing communities in the area. Pupils use an iLearn wiki to gather together fishing stories and SCRAN (a history resources site) to access newspaper archives and photographs. However, Jack also needs a photo sharing site as he wants pupils to take and comment on each others' photos and to upload scans of old photographs that they may have in their families.

Jack sends an email to a primary school teachers group, which he is a member of to see if anyone can recommend an appropriate system. Two teachers reply and both suggest that he uses KidsTakePics, a photo sharing site that allows teachers to check and moderate content. As KidsTakePics is not integrated with the iLearn authentication service, he sets up a teacher and a class account. He uses the iLearn setup service to add KidsTakePics to the services seen by the pupils in his class so that when they log in, they can immediately use the system to upload photos from their mobile devices and class computers.

Requirements elicitation (15/18)



● 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 elicitation (16/18)



- Uploading photos (iLearn)

- **Initial assumption:** A user or a group of users have one or more digital photographs to be uploaded to the picture sharing site. These are saved on either a tablet or laptop computer. They have successfully logged on to KidsTakePics.
- **Normal:** The user chooses upload photos and they are prompted to select the photos to be uploaded on their computer and to select the project name under which the photos will be stored. They should also be given the option of inputting keywords that should be associated with each uploaded photo. Uploaded photos are named by creating a conjunction of the user name with the filename of the photo on the local computer.
- On completion of the upload, the system automatically sends an email to the project moderator asking them to check new content and generates an on-screen message to the user that this has been done.

Requirements elicitation (17/18)



● Uploading photos (iLearn)

■ Initial assumption:

- A user or a group of users have one or more digital photographs to be uploaded to the picture sharing site. These are saved on either a tablet or laptop computer. They have successfully logged on to KidsTakePics.

■ Normal:

- The user chooses upload photos and they are prompted to select the photos to be uploaded on their computer and to select the project name under which the photos will be stored. They should also be given the option of inputting keywords that should be associated with each uploaded photo. Uploaded photos are named by creating a conjunction of the user name with the filename of the photo on the local computer.

Requirements elicitation (18/18)



● Uploading photos (iLearn) (cont.)

- On completion of the upload, the system automatically sends an email to the project moderator asking them to check new content and generates an on-screen message to the user that this has been done.
- What can go wrong:
 - No moderator is associated with the selected project.
 - Users should be informed that there could be a delay in making their photos visible.
 - Photos with the same name have already been uploaded by the same user.
 - The user should be asked if they wish to re-upload the photos with the same name, rename the photos or cancel the upload.
 - If they chose to re-upload the photos, the originals are overwritten.

Session Summary (1/4)



- 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.
- They often relate to the emergent properties of the system and therefore apply to the system as a whole.

Session Summary (2/4)



- The requirements engineering process is an iterative process that includes requirements elicitation, specification and validation.
- Requirements elicitation is an iterative process that can be represented as a spiral of activities – requirements discovery, requirements classification and organization, requirements negotiation and requirements documentation.
- You can use a range of techniques for requirements elicitation including interviews and ethnography. User stories and scenarios may be used to facilitate discussions.

