Faculty of ECMS / School of Computer Science

Software Engineering & Project Requirements Engineering

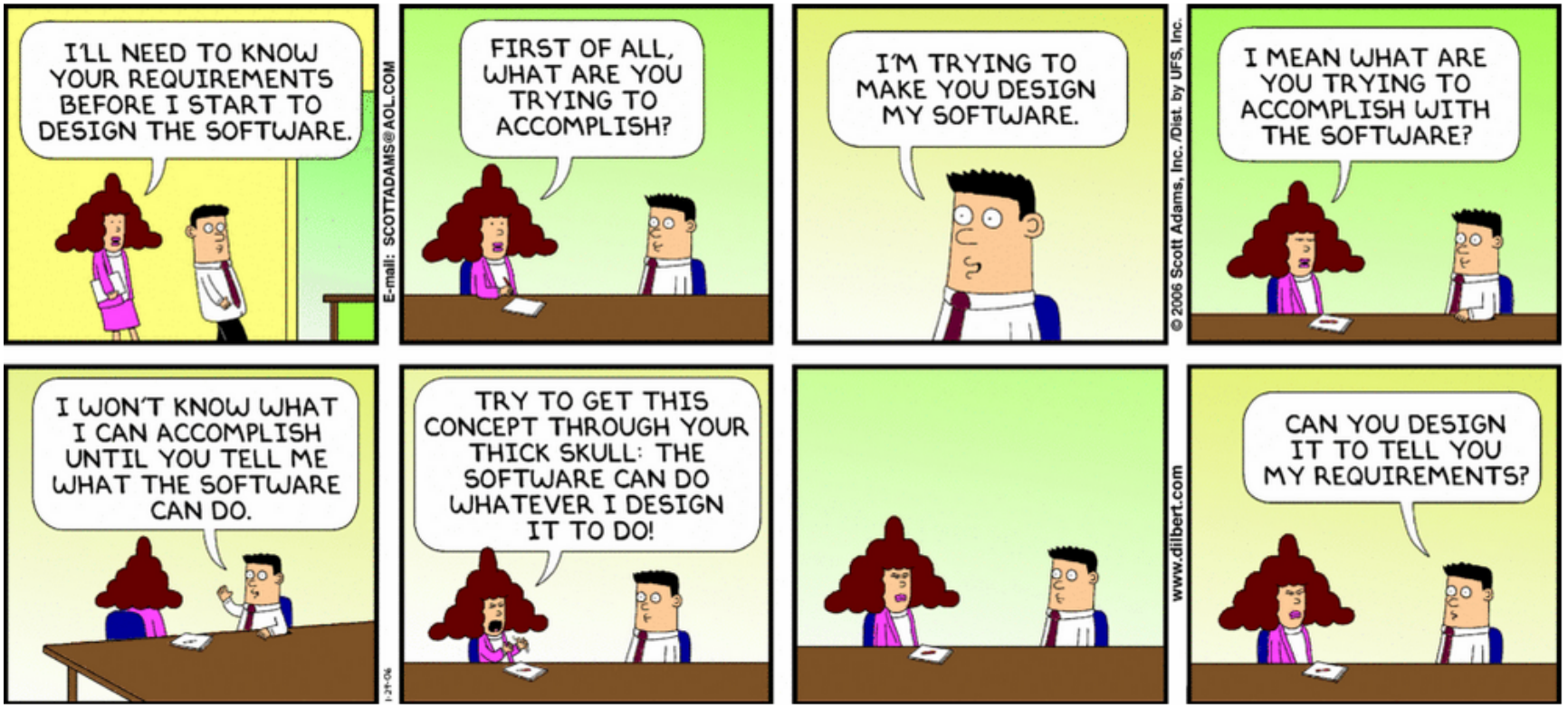
Project Management

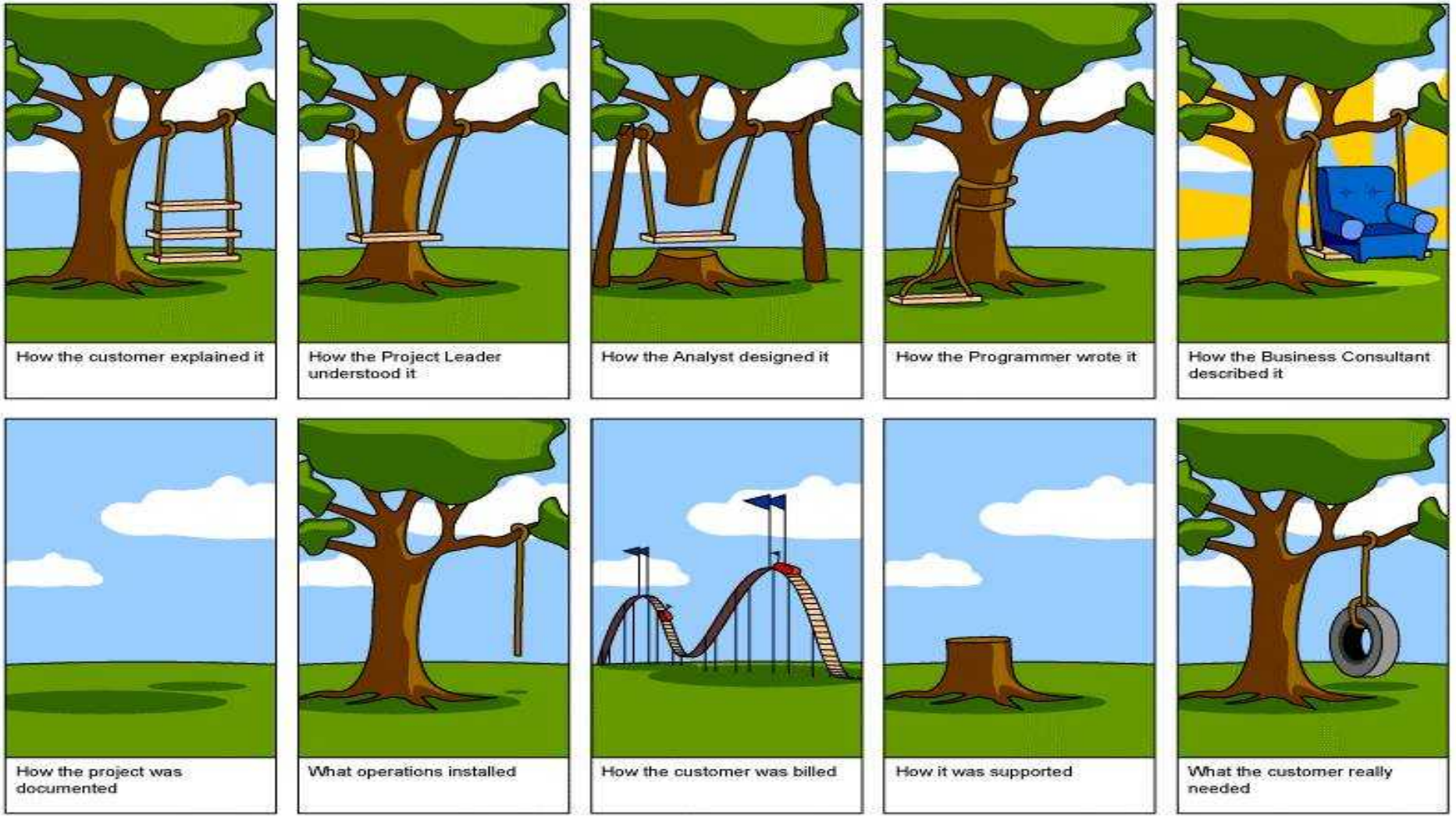
Lecture 3

Chapters 6, 7 (4 in Edition 9)
in the course text book

Questions

- What is requirements engineering?
- What are functional requirements?
- What are non-functional requirements?
- What are the techniques for requirements elicitation?
- How to document requirements?
- What activities are involved in requirements engineering?


I'LL NEED TO KNOW YOUR REQUIREMENTS BEFORE I START TO DESIGN THE SOFTWARE.
E-mail: SCOTTADAMS@AOL.COM
FIRST OF ALL, WHAT ARE YOU TRYING TO ACCOMPLISH?
I'M TRYING TO MAKE YOU DESIGN MY SOFTWARE.
© 2006 Scott Adams, Inc./Dist. by UFS, Inc.
I MEAN WHAT ARE YOU TRYING TO ACCOMPLISH WITH THE SOFTWARE?
I WON'T KNOW WHAT I CAN ACCOMPLISH UNTIL YOU TELL ME WHAT THE SOFTWARE CAN DO.
TRY TO GET THIS CONCEPT THROUGH YOUR THICK SKULL: THE SOFTWARE CAN DO WHATEVER I DESIGN IT TO DO!
www.dilbert.com
CAN YOU DESIGN IT TO TELL YOU MY REQUIREMENTS?


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

Outline

- Overview of Requirements Engineering
- Types of Requirements
- Techniques for Requirements Elicitation
- Requirements Documentation
- Requirements Management

What is Requirements Engineering?

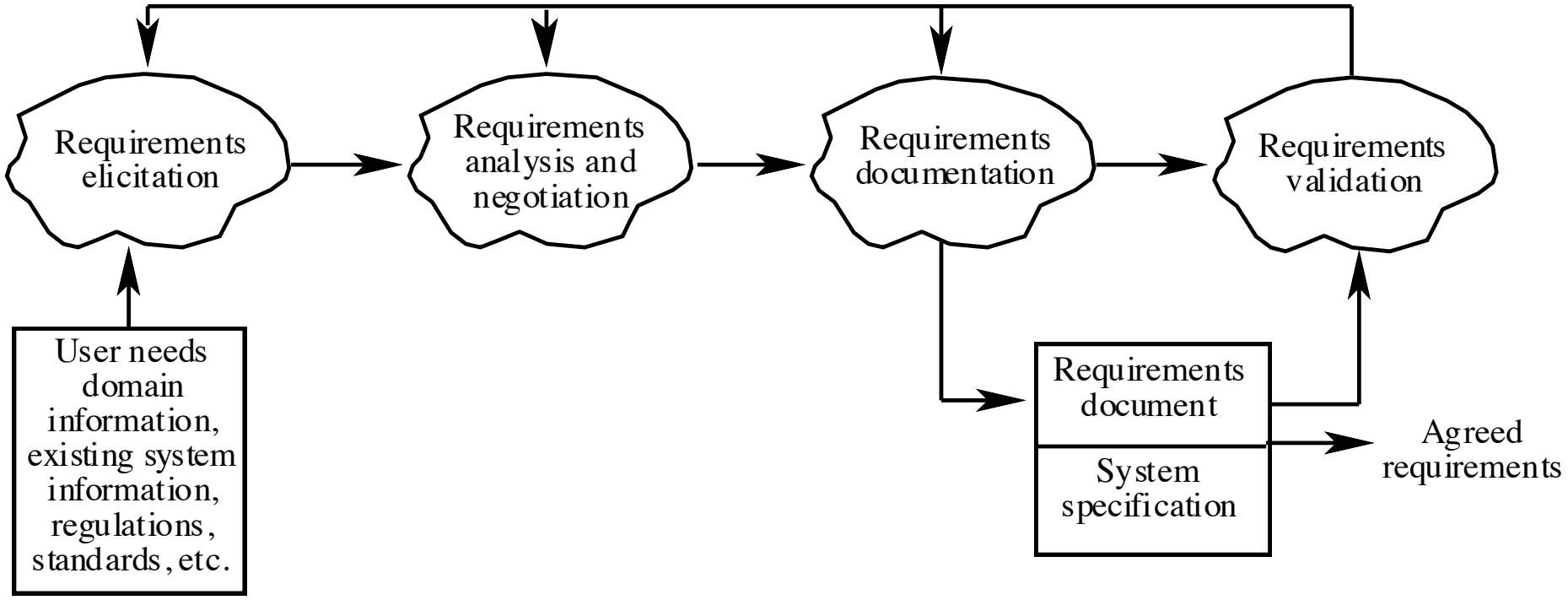
Requirements engineering is the branch of software engineering concerned with the real-world goals for, functions of, and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behaviour, and to their evolution over time and across software families.

[Zave 1997]

Requirements engineering is a systematic process of developing requirements through an iterative, co-operative process of analyzing the problem, documenting the resulting observations in a variety of representation formats and checking the accuracy of the understanding gained.

[Loucopoulos 1995]

Requirement Engineering Process



Types of Requirements

- User requirements
  - Statements in natural language plus diagrams of the services the system provides and its operational constraints. Written for e.g., customers.
  - Example: LIBSYS shall keep track of all data required by copyright licensing agencies in the UK and elsewhere.

- 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. Written for e.g., software architect/designers.
  - Example: on making a request for a document from LIBSYS, the requestor shall be presented with a form that records details of the user and the request made.

Types of Requirements (cont.)

- 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.
  - Defines *what* the system is supposed to 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.
  - Not directly related to specific functions.
  - Often related to system performance, usability, security, reliability etc.

Types of Requirements (cont.)

- Non-functional requirements generally are hard to verify
  - E.g., LIBSYS should be ease of use
  - E.g., LIBSYS should be highly available
- These vague goals cause problems to developers
  - Leave the scope for different interpretations and dispute after system delivery.
- So non-functional requirements should be quantitative, whenever it is possible.
  - E.g., A novice user should be able to use all functions of LIBSYS after a total of 5 days' training.
  - E.g., Downtime within normal working hours of LIBSYS shall not exceed five seconds in any one day

Look at These Requirement Examples

- Requirement 1:
  - A library system that provides a single interface to a number of databases of articles in different libraries. Users can search for, download and print these articles for personal study.

- Requirement 2:
  - The user interface for LIBSYS shall be implemented as simple HTML without frames or Java applets.
  - The system development process and deliverable documents shall conform to the process and deliverables defined in XYZCo-SP-STAN-95.

Outline

- Overview of Requirement Engineering
- Types of Requirements
- Techniques for Requirements Elicitation
- Requirements Documentation
- Requirements Management

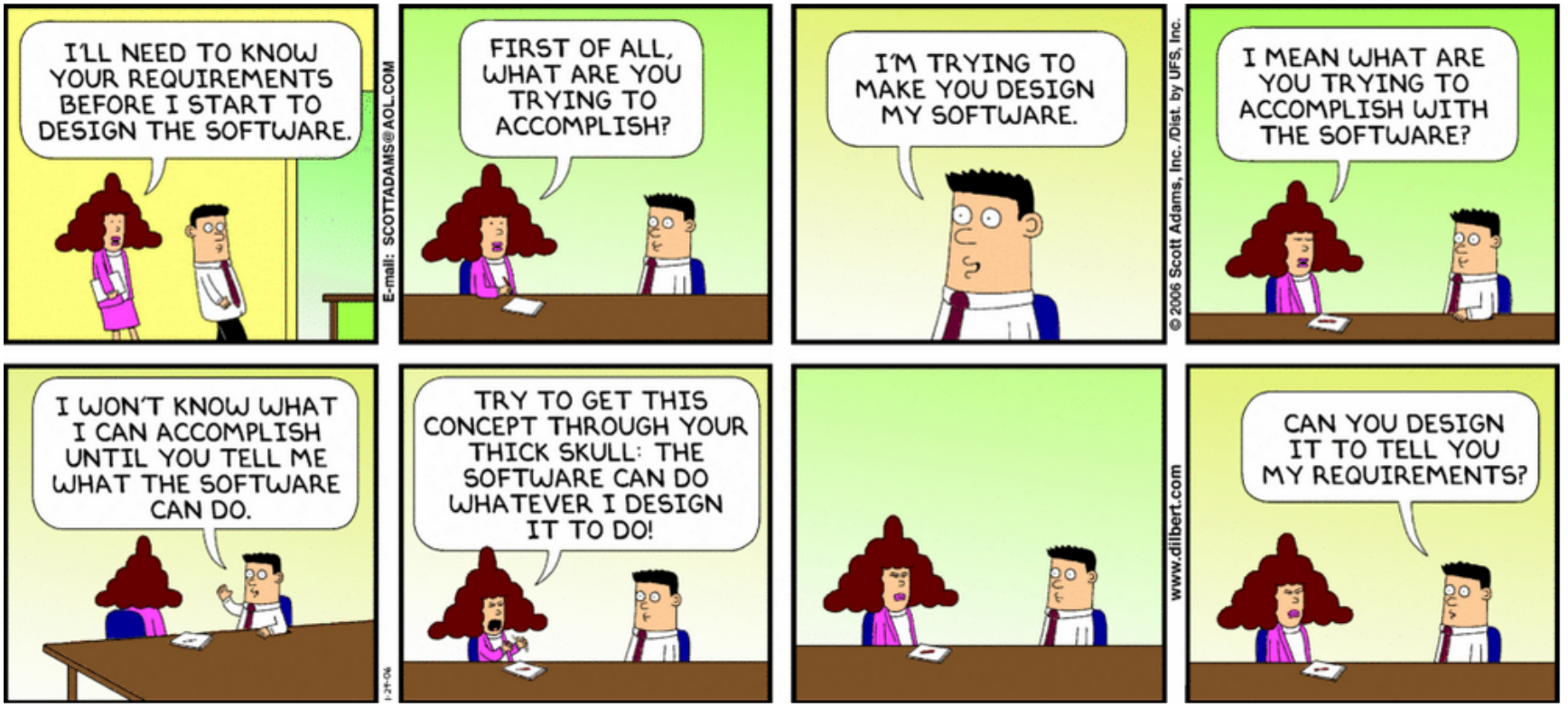
Requirements Elicitation

- Sometimes called 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.

Problems of Requirement Elicitation

- Stakeholders don't know what they really want.
- Stakeholders express requirements in their own terms.
- Different stakeholders may have conflicting requirements.
- Organisational 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.

Remember Dilbert...



Main Activities of Requirement Elicitation

- Requirements discovery
  - Interacting with stakeholders to discover their requirements.
- Requirements classification and organisation
  - Groups related requirements and organises them into coherent clusters.
- Prioritisation and negotiation
  - Prioritising requirements and resolving requirements conflicts.
- Requirements specification
  - Requirements are documented for system design or next round of requirement elicitation

Requirements Elicitation

- Techniques for requirements elicitation:
  - Interviews - the most commonly used technique.
  - Focus group
  - Ethnography - adopted from sociology and includes the observation of users.
  - Contextual query. It is a knowledge acquisition approach.
  - Use cases (Scenarios)
  - Goal-based techniques.
  - Qualify Function Deployment (QFD)
  - Prototype

The bottom four techniques can be used in the later stages of requirements elicitation.

Requirements Elicitation Techniques

Technique 1: Interview

- A technique for eliciting detailed information from stakeholders
- Type of interview:
  - Unstructured interviews (open interviews)
    - Ask more open-end questions, no predefined agenda
  - Structured interviews (closed interviews)
    - Ask more specific question (usually predefined) based on the analysis of the results from the unstructured interview – close end questions

Requirements Elicitation Techniques

Technique 1: Interview (Cont.)

- Consists of four phases:
  - Identifying candidates
  - Preparation
    - Arranging the interview
      - Schedule the interview in advance
      - Create a set of goals
      - Set the length of the interview
      - Request or give the interviewee the necessary materials
      - Confirm the interview 1 - 2 days in advance
    - Preparing the questions
      - Context-free product questions (open, close)
      - Context-specific product questions (open, close)
  - Conducting the interview
  - Follow-up

Requirements Elicitation Techniques

Technique 1: Interview (Cont.)

- Effective interviewers normally:
  - Are 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.

- **Good**: for getting an overall understanding of what stakeholders do & how they might interact with the system.
- **Not good**: for understanding domain requirements
  - Requirements engineers may not understand domain terminology;
  - Some domain knowledge is so familiar that people find it hard to articulate or think that it isn't worth articulating.

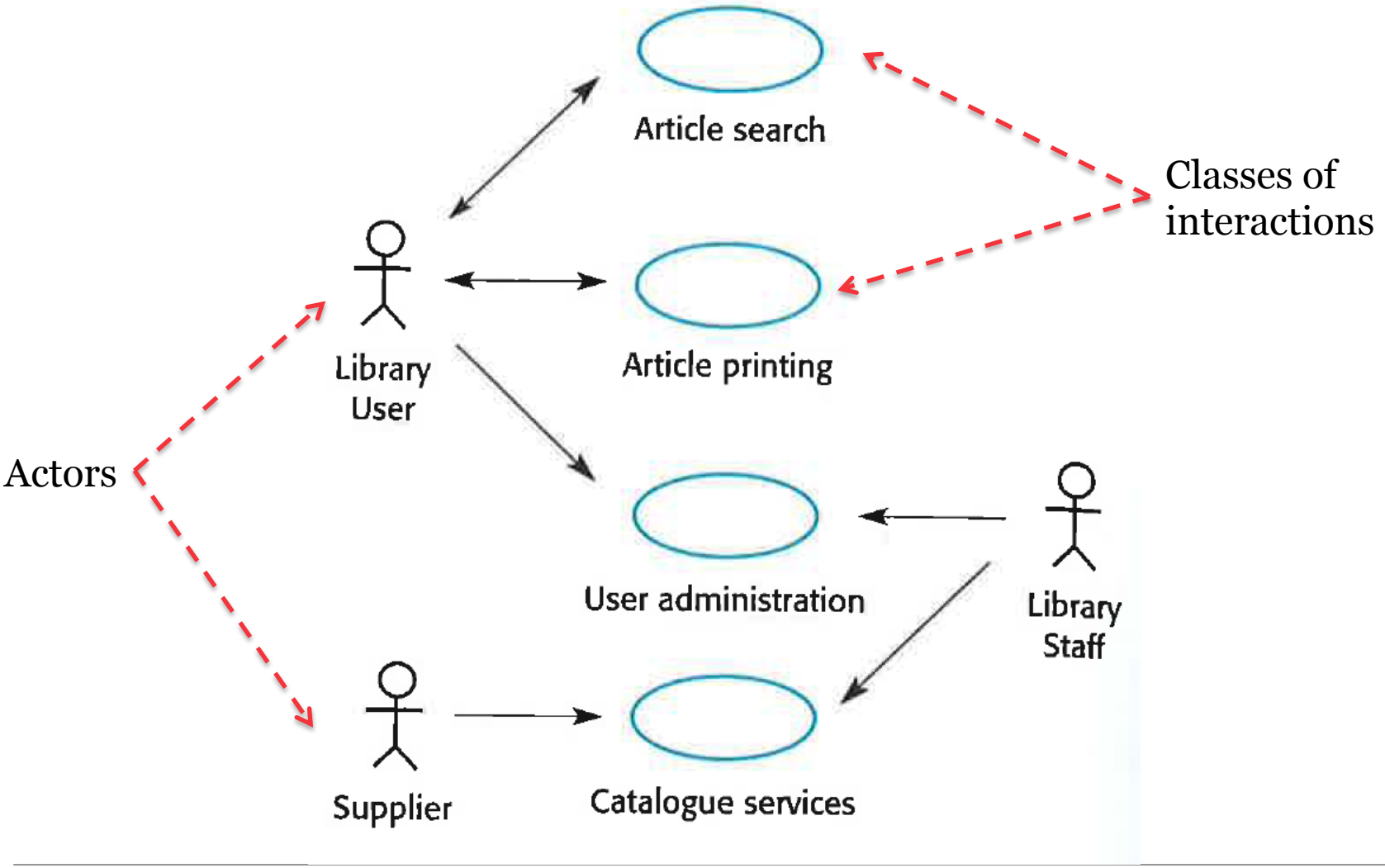
Requirements Elicitation Techniques

Technique 2: Use Case

- Use-cases are a scenario based technique which identify the **interactions** and the actors involved.
- A set of use cases should describe all possible interactions with the system.
- Part of UML
- Sequence diagrams may be used to add detail to use-cases by showing the sequence of event processing in the system.
- Focus on interactions, not effective to elicit constraints/ non-functional requirements.

Requirements Elicitation Techniques

Technique 2: Use Case (Cont.)



Requirements Elicitation Techniques

Technique 3: Prototype

- A prototype is an initial version of a system which is available early in the development phase
- Prototypes are valuable for requirements elicitation because users can experiment with the system and point out its strengths and weaknesses of the implemented requirements
- Rapid development of prototypes is essential so that they are available early in the elicitation process
  - Some functionality may be left out
  - Non-functional requirements (performance) are less stringent
  - No secondary functions (e.g. maintenance)
  - No complete documentation

Requirements Elicitation Techniques

Technique 3: Prototype (Cont.)

- The prototype allows users to experiment and discover what they really need to support their work; get their feedback
- Establishes feasibility and usefulness before high development costs are incurred
- Essential for developing the 'look and feel' of a **user interface**
  - Probably the only technique to validate interface requirements
- Forces a detailed study of the requirements which reveals inconsistencies and omissions

Requirements Elicitation Techniques

Technique 3: Prototype (Cont.)

- Paper Prototyping
  - A paper **mock-up** of the system for feedback
- "Wizard of Oz" Prototyping
  - A person simulates the responses of the system in response to some user inputs
  - http://en.wikipedia.org/wiki/Wizard_of_Oz_experiment
- Executable Prototyping
  - Some rapid development language/environments are used to develop an executable prototype

Requirements Elicitation Techniques

Technique 3: Prototype (Cont.)

Next time

- Overview of Requirement Engineering
- Types of Requirements
- Techniques for Requirements Elicitation
- Requirements Documentation
- Requirements Management