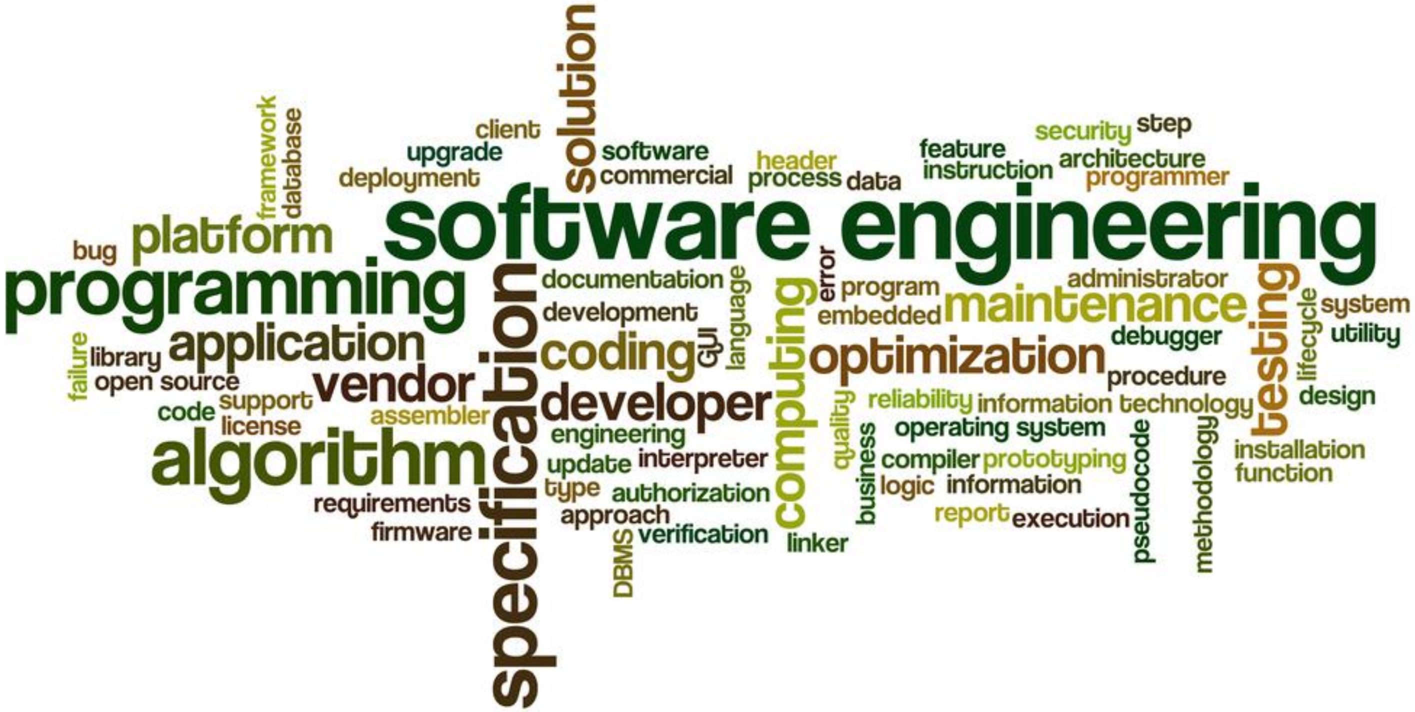


SWE 205: Introduction to Software Engineering

Lecture 7

Requirements Engineering II

Course Topics

- ~~Introduction~~
- ~~Software Process Models~~
- Requirements Engineering
- Modeling
- Programming Languages
- Software Construction Techniques
- Testing
- Project Management
- Refactoring
- Ethical Issues

Lecture Objectives

✓ Requirements Engineering Process activities

1 - Feasibility study
2 - Requirements elicitation and analysis
3 - Requirements specification
4 - Requirements validation

✓ Elicitation and analysis

- Interviewing, workshops, … etc.
- scenarios
- use cases

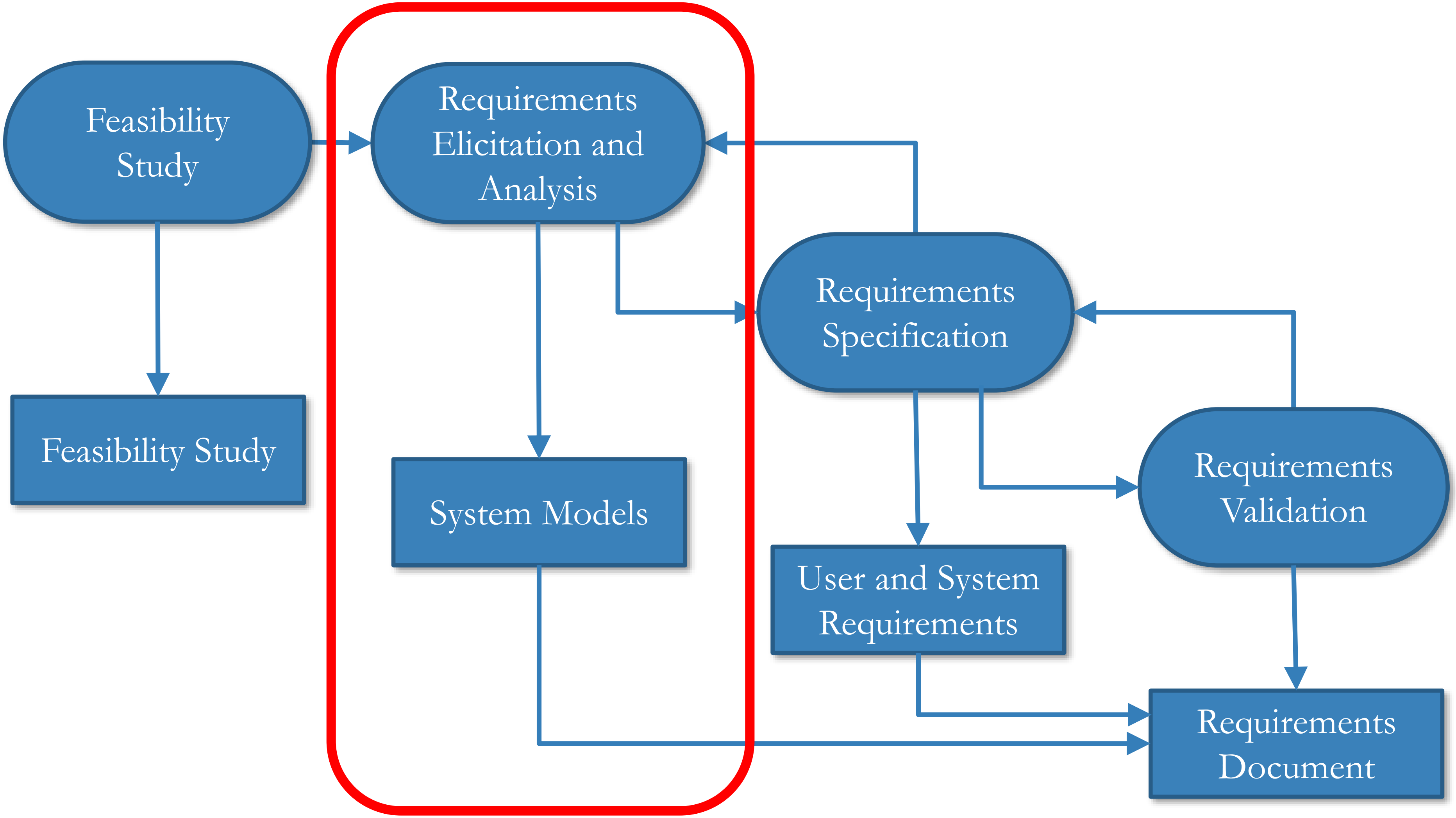
Requirements Engineering Processes

- The processes used for RE vary widely depending on the **application domain**, the **people** involved and the **organisation** developing the requirements.

- However, there are a number of **generic activities** common to all processes

1 - Feasibility study
2 - Requirements elicitation and analysis;
3 - Requirements specification;
4 - Requirements validation;

The requirements engineering process


Feasibility Study
Feasibility Study
Requirements Elicitation and Analysis
System Models
Requirements Specification
User and System Requirements
Requirements Validation
Requirements Document

Elicitation and analysis

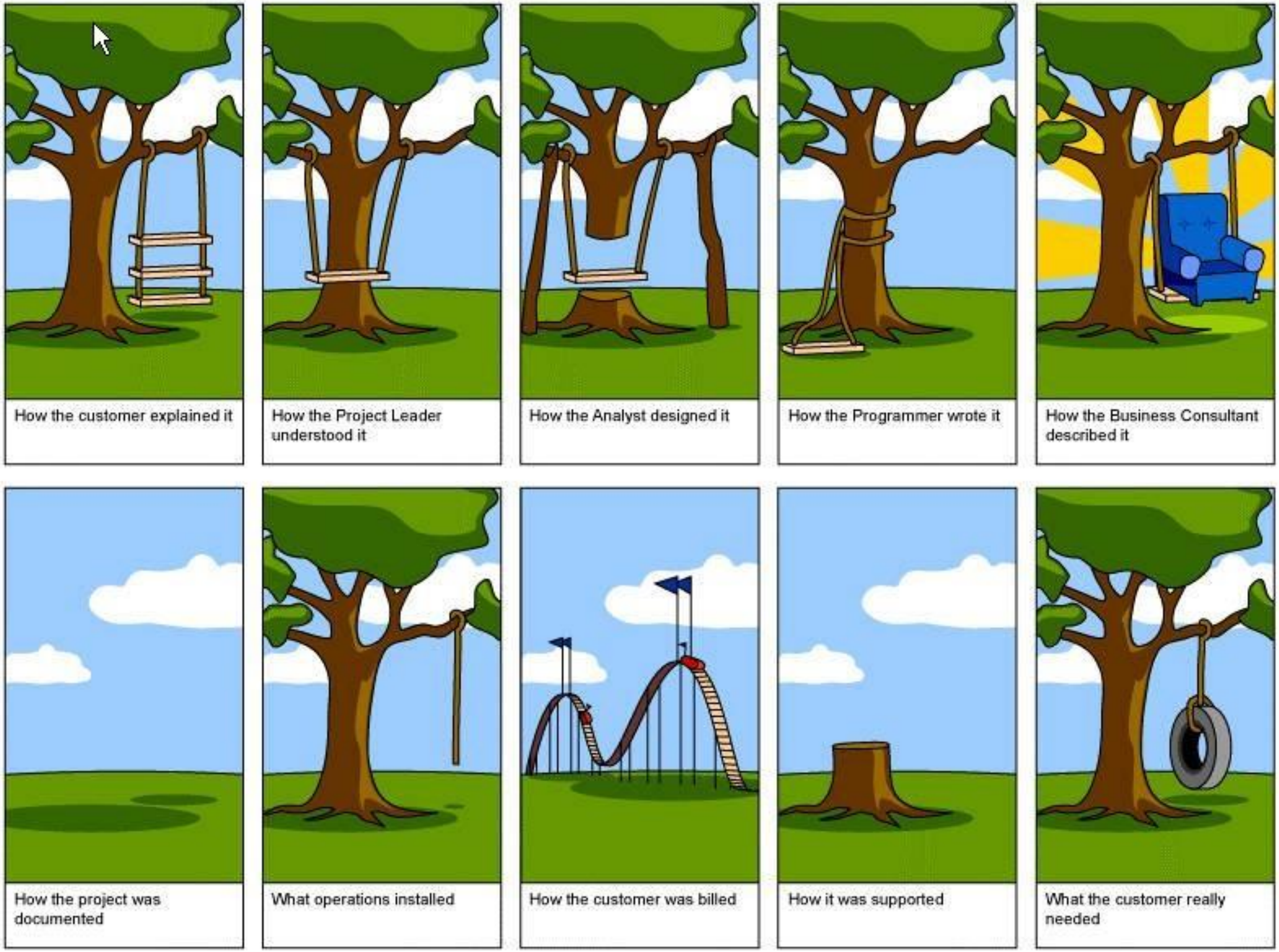
- Sometimes called
  - Requirements **elicitation**
  - Requirements **discovery**

- 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, etc. These are called ***stakeholders***.

Problems of requirements analysis

- Issues that might be faced:
  - 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 change.



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

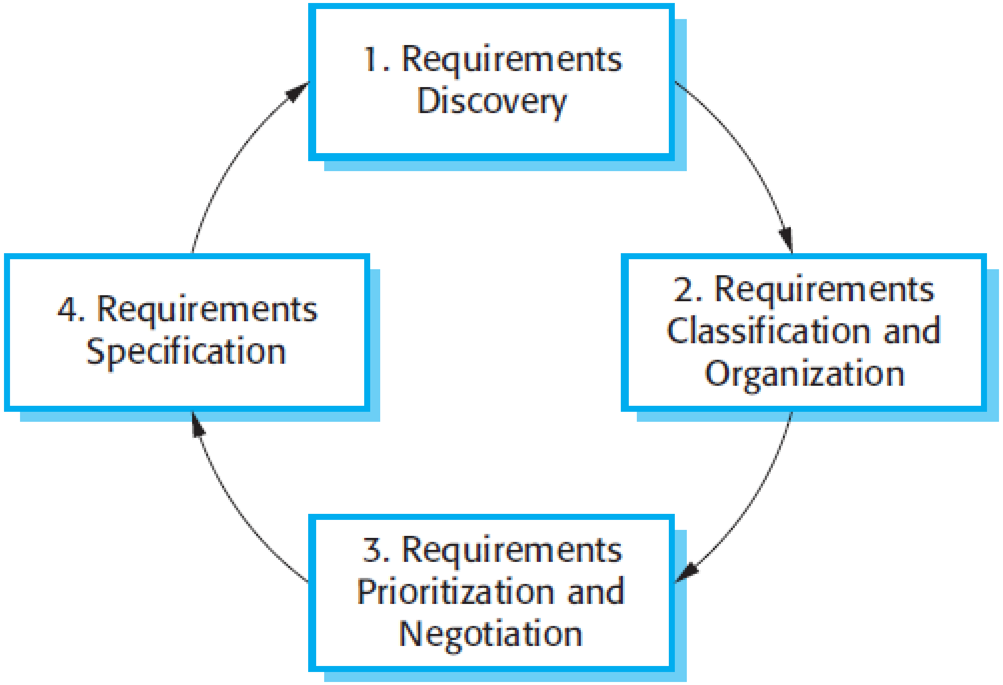
What operations installed

How the customer was billed

How it was supported

What the customer really needed

Elicitation & Analysis > Process activities


1. Requirements Discovery
2. Requirements Classification and Organization
3. Requirements Prioritization and Negotiation
4. Requirements Specification

Elicitation & Analysis > Process activities

1 - Requirements discovery

- Interacting with stakeholders to discover their requirements. Domain requirements are also discovered at this stage.

2 - Requirements classification and organisation

- Groups related requirements and organises them into coherent clusters.

3 - Requirements prioritisation and negotiation

- Prioritising requirements and resolving requirements conflicts.

4 - Requirements documentation

- Requirements are documented and input into the next round of the spiral.

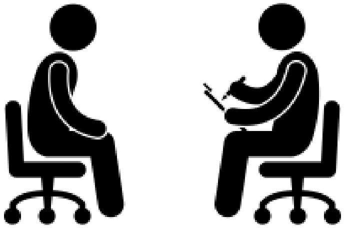
Requirements discovery

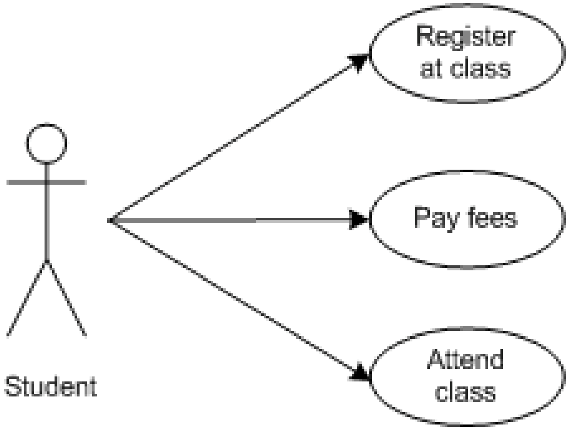
- The process of **gathering information** about the proposed and existing systems and distilling the user and system requirements from this information.

- Sources of information include:
  - Documentation
  - System stakeholders
  - Specifications of similar systems

Eliciting Requirements Techniques

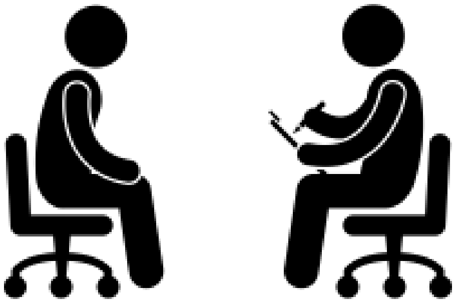
- Interviews
- Workshops
- Scenarios
- Use Cases
- Ethnography





Interviews

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



- Open Vs. Closed Interviews

- Context free and solution context questions

- 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.

Workshop

- Meetings are conducted and attended by both software engineers and customers
- Rules for preparation and participation are established
- An agenda is suggested
- A "**facilitator**" (can be a customer, a developer, or an outsider) controls the meeting
- A "**definition mechanism**" (can be work sheets, flip charts, or wall stickers or an electronic bulletin board, chat room or virtual forum) is used
  - The goal is
  - to identify the problem
  - propose elements of the solution
  - negotiate different approaches, and
  - specify a preliminary set of solution requirements

Ethnography

- A social scientist spends a considerable time observing and analysing how people actually work.
- People do not have to explain or articulate their work.

- Social and organisational factors of importance may be observed.
- Ethnographic studies have shown that work is usually richer and more complex than suggested by simple system models.

Scope of ethnography

- Requirements that are derived from the way that people actually work rather than the way in which process definitions suggest that they ought to work.

- Requirements that are derived from cooperation and awareness of other people's activities.
  - Awareness of what other people are doing leads to changes in the ways in which we do things.

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

Scenarios

- Scenarios are like **stories**

- Or real-life examples of how a system can be used.

- They 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.

LIBSYS scenario (1)

Initial assumption: The user has logged on to the LIBSYS system and has located the journal containing the copy of the article.

Normal: The user selects the article to be copied. He or she is then prompted by the system to either provide subscriber information for the journal or to indicate how they will pay for the article. Alternative payment methods are by credit card or by quoting an organisational account number.

The user is then asked to fill in a copyright form that maintains details of the transaction and they then submit this to the LIBSYS system.

The copyright form is checked and, if OK, the PDF version of the article is downloaded to the LIBSYS working area on the user's computer and the user is informed that it is available. The user is asked to select a printer and a copy of the article is printed. If the article has been flagged as 'print-only' it is deleted from the user's system once the user has confirmed that printing is complete.

LIBSYS scenario (2)

What can go wrong: The user may fail to fill in the copyright form correctly. In this case, the form should be re-presented to the user for correction. If the resubmitted form is still incorrect then the user's request for the article is rejected.

The payment may be rejected by the system. The user's request for the article is rejected.

The article download may fail. Retry until successful or the user terminates the session.

It may not be possible to print the article. If the article is not flagged as 'print-only' then it is held in the LIBSYS workspace. Otherwise, the article is deleted and the user's account credited with the cost of the article.

Other activities: Simultaneous downloads of other articles.

System state on completion: User is logged on. The downloaded article has been deleted from LIBSYS workspace if it has been flagged as print-only.

Use cases

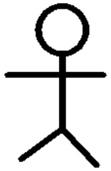
- Use-cases are a scenario based technique in the **UML** which identify the actors in an interaction and which describe the interaction itself.

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

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

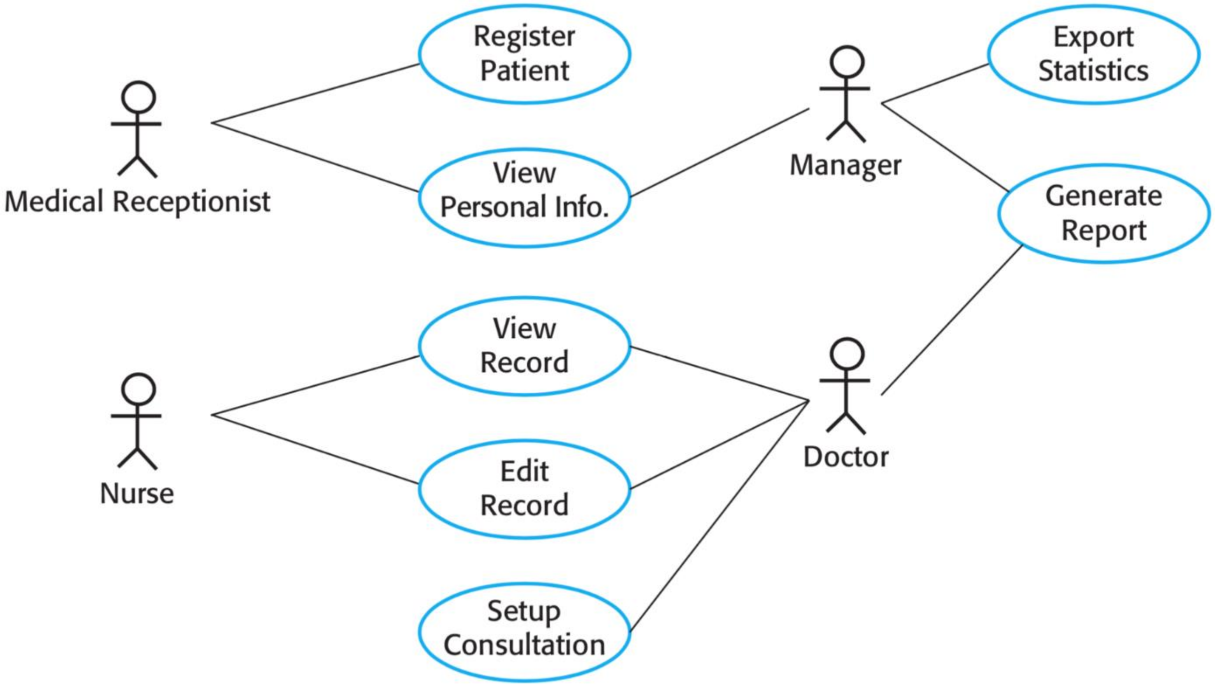
Use cases + Example

- Two Symbols:
  - Actor



  - Use case

Example



Use case description

- A description of **events** needed to accomplish the required behavior of the use case.

- **WHAT** the system should do, **not HOW** the system does it.

- Written in the language of the domain, not in terms of implementation.

Use case description – Flow of Events :

- When and how the use case starts and ends
- What interaction the use case has with the actors
- What data is needed by the use case
- The normal sequence of events for the use case
- The description of any alternate or exceptional flows

Use case description – Flow of Events :

- Basic Flow of Event
- Alternate Flow of Event

Examples: Ticket Booking

Primary Flow

The steps for the primary flow of events include:

1. The use case begins when the customer selects the option to view flight information.
2. The system prompts for the departure and destination cities and the departure and return dates.
3. The user enters the departure and destination city, departure date, and return date.
4. The system displays a list of available flights, including the fare.
   A1: There are no available flights.
5. The user selects the flight they would like to reserve.
6. The system displays all available fare options for that flight.
7. The user selects the fare option they would like to reserve.
   A2: The user selects a free ticket through frequent−flyer membership.

Ticket Booking …

8. The system displays the fare that the user will pay.
9. The user confirms the rate.
10. The system prompts for a credit card type, number, name, and expiration date.
11. The user enters the card type, number, name, and expiration date.
12. The system submits the credit purchase.

    A3: Account not found

    A4: Insufficient funds

    E1: Credit system not accessible
13. The system reserves a seat on the plane for the user.
14. The system generates and displays a confirmation code to the user.
15. The user confirms receipt of the code.
16. The use case ends.

Ticket Booking….

Alternate Flows

A1: No available flights

1. The system displays a message that there are no available flights for the departure and destination cities, departure date, and return date entered.
2. The user confirms the message.
3. The flow returns to the primary flow, step 2.

A2: Free ticket through frequent−flyer membership

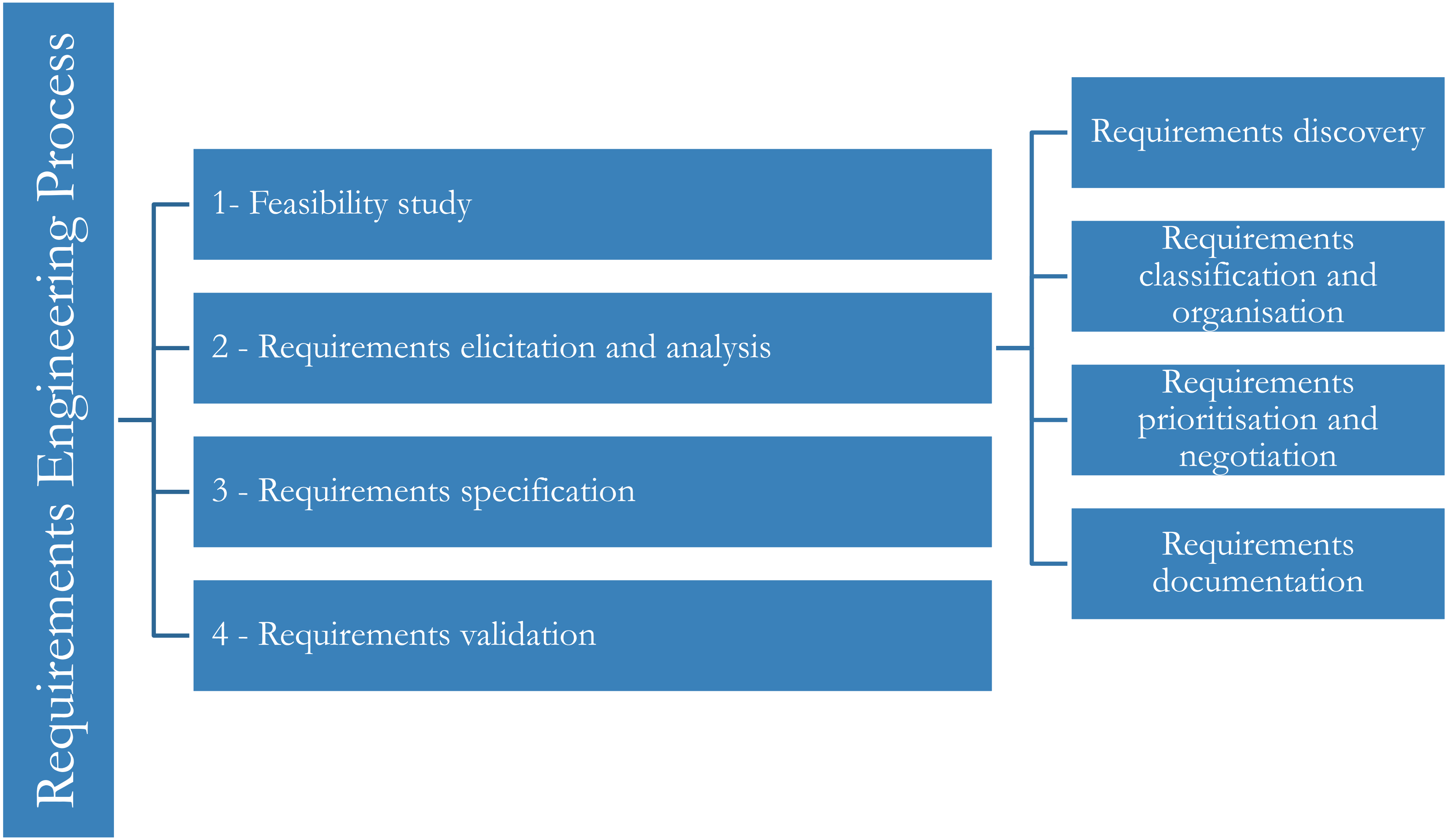
1. The system prompts for the frequent−flyer number.

…………..

Other Elicitation Techniques

- Brainstorming
- Questionnaire
- …

Summary



Requirements Engineering Process

1- Feasibility study

2 - Requirements elicitation and analysis

3 - Requirements specification

4 - Requirements validation

Requirements discovery

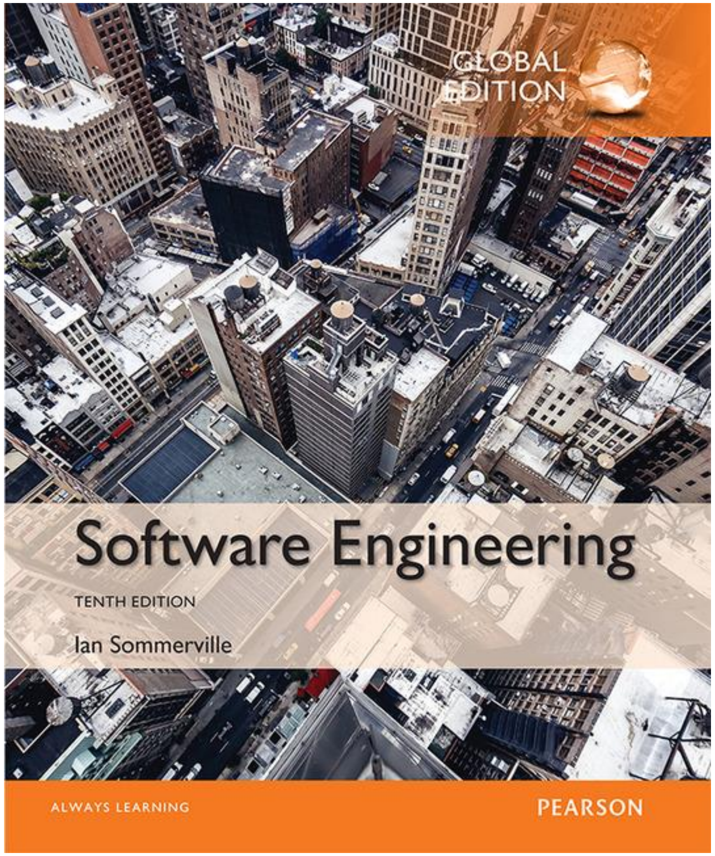
Requirements classification and organisation

Requirements prioritisation and negotiation

Requirements documentation

Read

Chapter 4



References

- Ian Sommerville, “Software Engineering”, 10th Edition, Addison-Wesley, 2015.
- Timothy C. Lethbridge and Robert Laganière, “Object-Oriented Software Engineering: Practical Software Development using UML and Java”, 2nd Edition, McGraw Hill, 2001.
- R. S. Pressman, Software Engineering: A Practitioner’s Approach, 10th Edition, McGraw-Hill, 2005.