

---

# Software Architecture

## Laboratory 03

### Requirements Engineering

Néstor Cataño  
Innopolis University

Spring 2016

# What's Requirements Engineering

---

**“Requirements engineering is a process that involves all the activities required to create and maintain a system requirement document”**

**Based on chapter 4 of Software Engineering, by Ian Sommerville**

# What's Requirements Engineering

---

## Web

Requirement Engineering, also called requirement analysis, is the process of determining **user expectations** for a new modified product.

These features, called requirements, must be **quantifiable**, **relevant**, and **detailed**.

Requirements Engineering involves frequent communication with **stakeholders** to determine system expectations, and conflict resolution.

# Requirements Engineering

---

- A recent **European survey** showed that the principal areas in software development and production are the **requirements specification** and the **management of user requirements**.
- Standards such as the **ISO 9000** have emphasised the importance of process definition and standardization. Surprisingly, however, **requirements engineering has been neglected** in the move towards process definition and standardisation.

# Errors in Software

---

- The development of computer-based systems has been plagued with bugs since the 1960s.
- Systems may be delivered late, over budget, they do not do what users really want and they are often never used their full effectiveness by the people who have paid for them.

**“Most of these problems are related with software requirements”**

## Errors arise when ...

---

- The requirements do not reflect the real needs of the customer of the system.
  - Requirements are **inconsistent** and/or **incomplete**
  - It's **expensive** to make changes to the requirements once they have been agreed.
  - There are **misunderstandings** between customers, those developing the system requirements and software engineers developing or maintaining the system.
-

## Best way to reduce errors is

---

- to **improve** the process of discovering, understanding, negotiating, describing, validating and managing system requirements.
- to do it in a **gradual way** where one introduces improved procedures over a period of time.

**The business risks of major process reengineering are simply too high to be acceptable**

# Software Requirements

---

- They are **descriptions** of how a system should **behave**.
- They are defined during the **early stages** of a system development as a **specification** of what should be implemented.



# Requirements describe

---

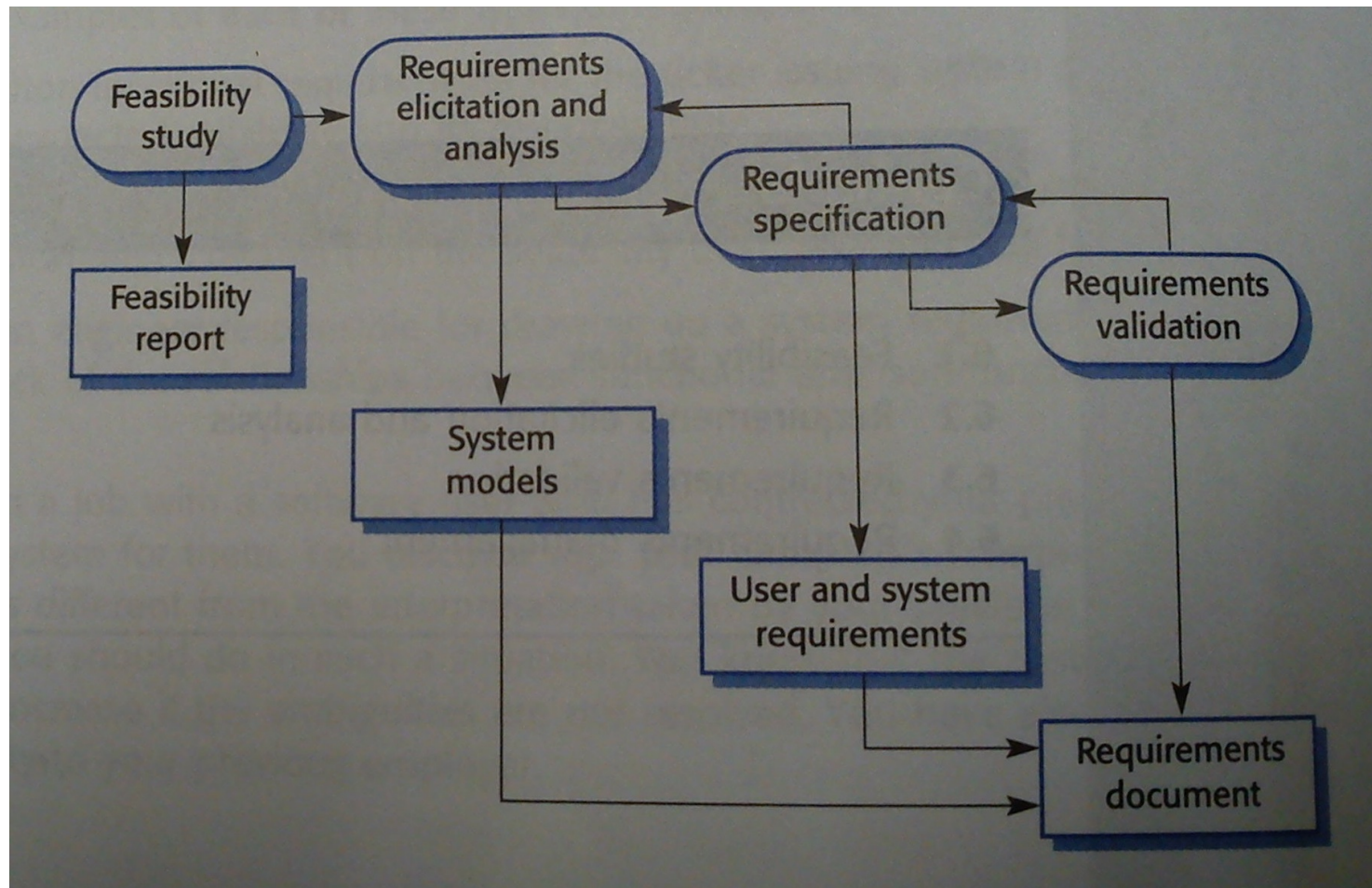
- **a user-level facility**: the word processor must include a spell checker.
  - **a general system property**: the system must ensure that personal information is never made available without authorisation
  - **a specific constraint about the system**: the sensor must be polled 10 times per second.
  - **a constraint about the development of the system**: the system must be developed using C language.
-

# Stakeholders

---

- System **stakeholders are people** who will be affected by the system and who have a direct or indirect influence on the system requirements.
  - e.g. **end-users, managers, engineers, customers, external bodies.**

# Requirements Engineering Processes



# Requirements Engineering Processes

---

1. **Feasibility Study**. Is the system feasible?
2. **Requirements Elicitation**. The system requirements are discovered through consultation with stakeholders, from systems documents, domain knowledge and market studies.
3. **Requirements Analysis**. The requirements are analysed in detail, and should be some formal negotiation process involving different stakeholders to decide on which requirements are to be accepted.
4. **Requirements Validation**. These should be a careful check of the requirements for consistency and completeness.
5. **Requirements Management**. Requirements can change so people need to develop a better/different understanding of the system.

# Feasibility Study

---

Description of the system and  
how it will be used within the  
organisation



Feasibility  
Study



Feasibility Report

---

# Feasibility Study

---

1. Does the system contribute to the overall objectives of the organization?
2. Can the system be implemented using current technology and within given cost and schedule constraints?
3. Can the system be integrated with other systems which are already in place?

**“If a system does not support these objectives, it has no real value to the business.”**

# Carrying out a Feasibility Study involves

---

1. Information Assessment
2. Information Collection
3. Report Writing

# Information Assessment

---

- **Information sources are identified**
  - **managers** of departments where the system will be used,
  - **software engineers** that are familiar with the type of system that is proposed,
  - **technology experts**,
  - **end-users** of the system



# Information Collection – The sources are interviewed

---

1. How would the organization **cope** if the system was not implemented?
2. What are the **problems** with current processes and how would a new system help **alleviate** these problems?
3. What **direct contribution** will the system make to the business objectives?
4. Can **information** be **transferred** to and from organizational systems?
5. Does the system **require technology** which has not previously been used in the organization?
6. What **must be supported** by the system and what **need not be supported**?

# Report Writing

---

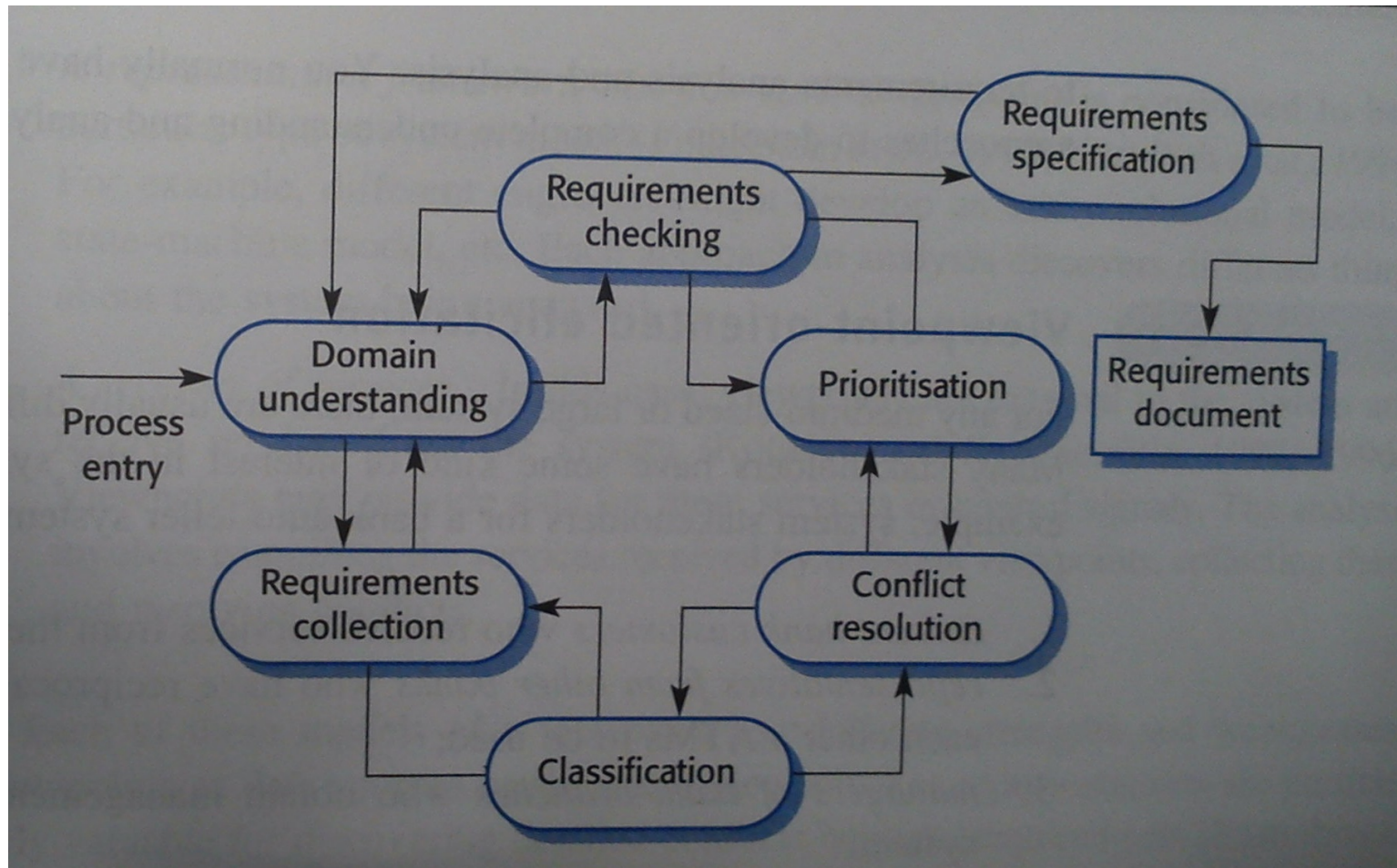
- When the information is gathered, a feasibility report is prepared.
- This should make a recommendation whether or not the system development should continue

# Elicitation and Analysis are difficult

---

1. **Stakeholders** often **don't know what they want** from the computer system except in the most general terms
2. **Stakeholders** might find **difficult to articulate** what they want from the system.
3. **Stakeholders** may make **unrealistic demands** about the system because they are unaware of the cost their requests
4. Different **Stakeholders** have **different requirements** and they may express these in different ways.
5. **Political factors** may influence the requirements of the system. These may come from managers who demand specific system requirements because these allow them to increase their influence in the organization.
6. Requirements may change. New requirements may emerge from **new Stakeholders** who were not originally consulted.

# Elicitation and Analysis processes



# Elicitation and Analysis processes

---

1. **Domain Understanding.** Analysts must develop their understanding of the application domain. For example, if a system for a supermarket is required, the analyst must find out how supermarkets operate.
2. **Requirements Collection.** This is the process of interacting with stakeholders in the system to discover their requirements.
3. **Classification.** It takes the unstructured collection of requirements and organizes them into coherent clusters.
4. **Conflict Resolution.** Finding and resolving requirements conflicts
5. **Prioritization.** Interacting with stakeholders to discover the most important requirements.
6. **Requirements Checking.** Requirements are checked to discover if they are **complete**, **consistent**, and in accordance with what stakeholders really want from the system.