# Software Engineering

By B. Pranalini

# Unit 2 Part 2: Requirements Engineering Processes

- Processes used to discover, analyse and validate system requirements

# Objectives

- To describe the principal requirements engineering activities
- To introduce techniques for requirements elicitation and analysis
- To describe requirements validation
- To discuss the role of requirements management in support of other requirements engineering processes

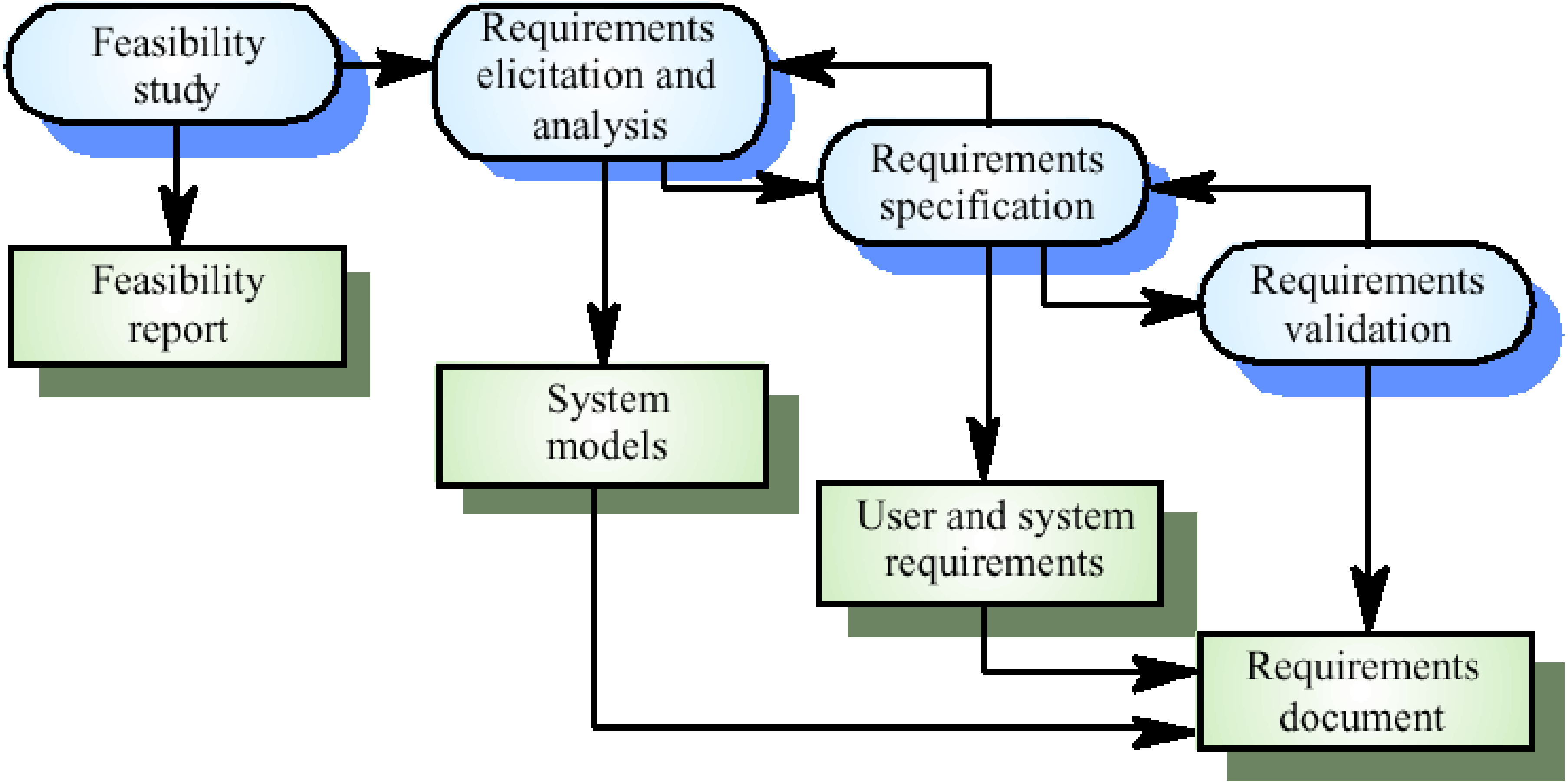
# Topics covered

- Feasibility studies
- Requirements elicitation and analysis
- Requirements validation
- Requirements management

# 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
  - Requirements elicitation
  - Requirements analysis
  - Requirements validation
  - Requirements management

# The requirements engineering process



# Feasibility studies

- A feasibility study decides whether or not the proposed system is worthwhile
- A short focused study that checks
  - If the system contributes to organisational objectives
  - If the system can be engineered using current technology and within budget
  - If the system can be integrated with other systems that are used

# Feasibility study implementation

- Based on information assessment (what is required), information collection and report writing
- Questions for people in the organisation
  - What if the system wasn't implemented?
  - What are current process problems?
  - How will the proposed system help?
  - What will be the integration problems?
  - Is new technology needed? What skills?
  - What facilities must be supported by the proposed system?

# REQUIREMENTS ELICITATION AND ANALYSIS

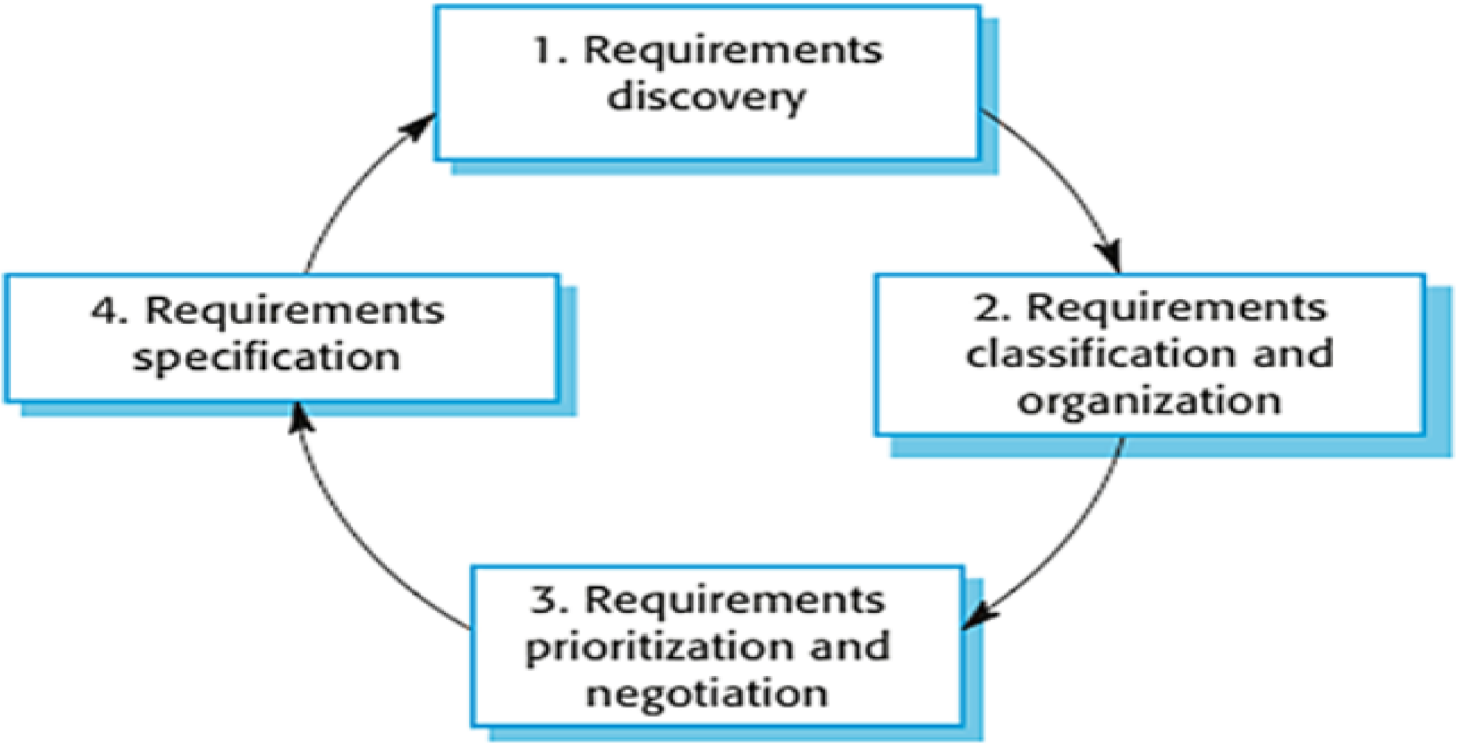
- It’s a process of interacting with customers and end-users to find out about the domain requirements, what services the system should provide, and the other constrains.
- *Domain requirements reflect the environment in which the system operates so, when we talk about an application domain we mean environments such as train operation, medical records, e-commerce etc.*
- It may also involve a different kinds of stockholders; end-users, managers, system engineers, test engineers, maintenance engineers, etc.
- *A stakeholder is anyone who has direct or indirect influence on the requirements.*

# Problems of requirements analysis

- 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

**The requirements elicitation and analysis has 4 main process**

- We typically start by gathering the requirements, this could be done through a general discussion or interviews with your stakeholders, also it may involve some graphical notation.
- Then you organize the related requirements into sub components and prioritize them, and finally, you refine them by removing any ambiguous requirements that may raise from some conflicts.
- Here are the 4 main process of requirements elicitation and analysis.


1. Requirements discovery
2. Requirements classification and organization
3. Requirements prioritization and negotiation
4. Requirements specification

The process of requirements elicitation and analysis

It shows that it’s an iterative process with a feedback from each activity to another. The process cycle starts with requirements discovery and ends with the requirements document. The cycle ends when the requirements document is complete.

**1. Requirements Discovery**

- It is also called requirements elicitation.
- It’s the process of interacting with, and gathering the requirements from, the stakeholders about the required system and the existing system (if exist).
- It can be done using some techniques, like interviews, scenarios, prototypes, etc, which help the stockholders to understand what the system will be like.
- Stakeholders range from end-users of a system through managers to external stakeholders such as regulators, who certify the acceptability of the system. For example, system stakeholders for the mental healthcare patient information system include:
  1. Patients whose information is recorded in the system.
  2. Doctors who are responsible for assessing and treating patients.
  3. etc

- These different requirements sources (stakeholders, domain, systems) can all be represented as system viewpoints with each viewpoint showing a subset of the requirements for the system.
- Different viewpoints on a problem see the problem in different ways.
- However, their perspectives are not completely independent but usually overlap so that they have common requirements.
- We can use these viewpoints to structure both the discovery and the documentation of the system requirements.

## Interviews

- In Interviews, requirements engineering teams put the questions to the stakeholder about the system that's currently used, and the system to be developed, and hence they can gather the requirements from the answers.
- The questions fall under two categories:
  - *Closed-Ended* questions: A pre-defined set of question.
  - *Open-Ended* questions: There is no a pre-defined expected answer, they are more of generic questions. It's used to explore issues that's not clear in a less structured way.
- In practice, interviews usually use mixture of both. Usually, start with the open-ended questions, and then use the closed-ended questions to be more specific about some requirements that aren't clear yet.

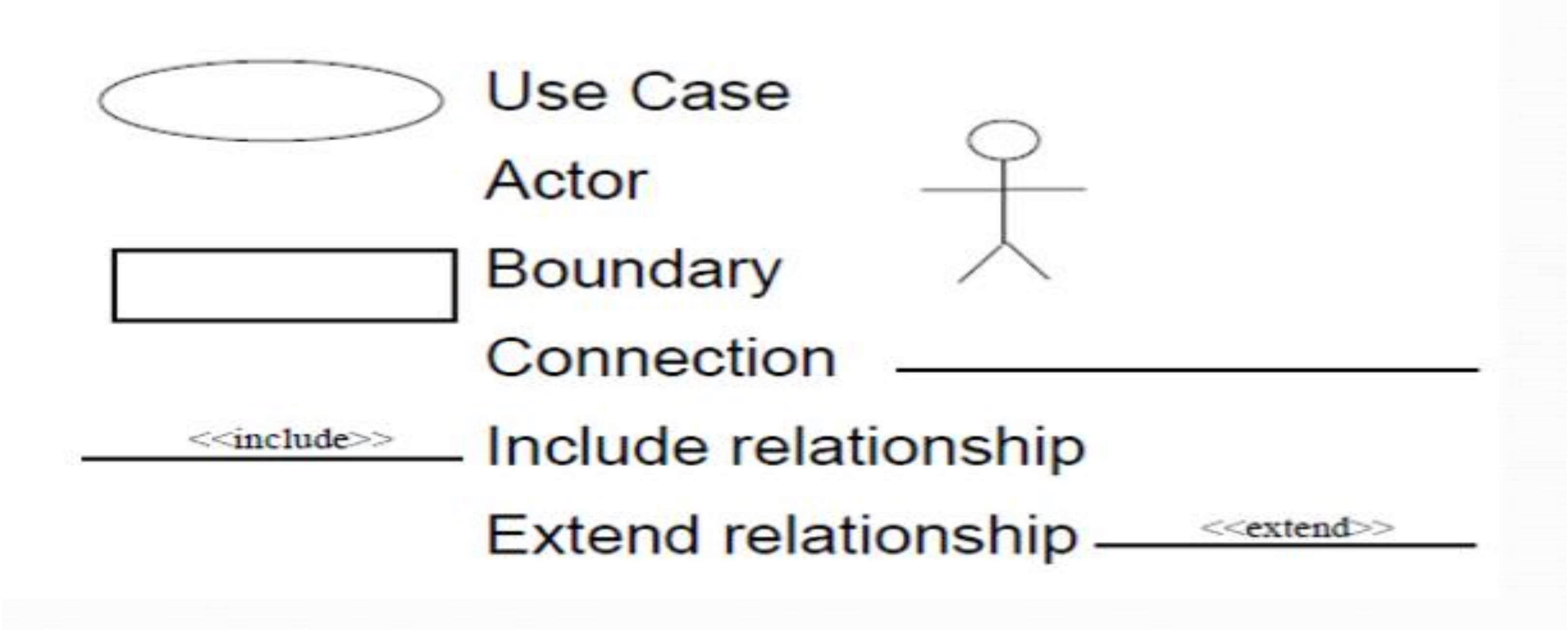
- Interviews are good to get an overall understanding of what stakeholders need, how they might interact with the new system, and the difficulties they face with the current system.
- However, interviews aren’t so helpful in understanding the domain requirements. This is for two reasons:

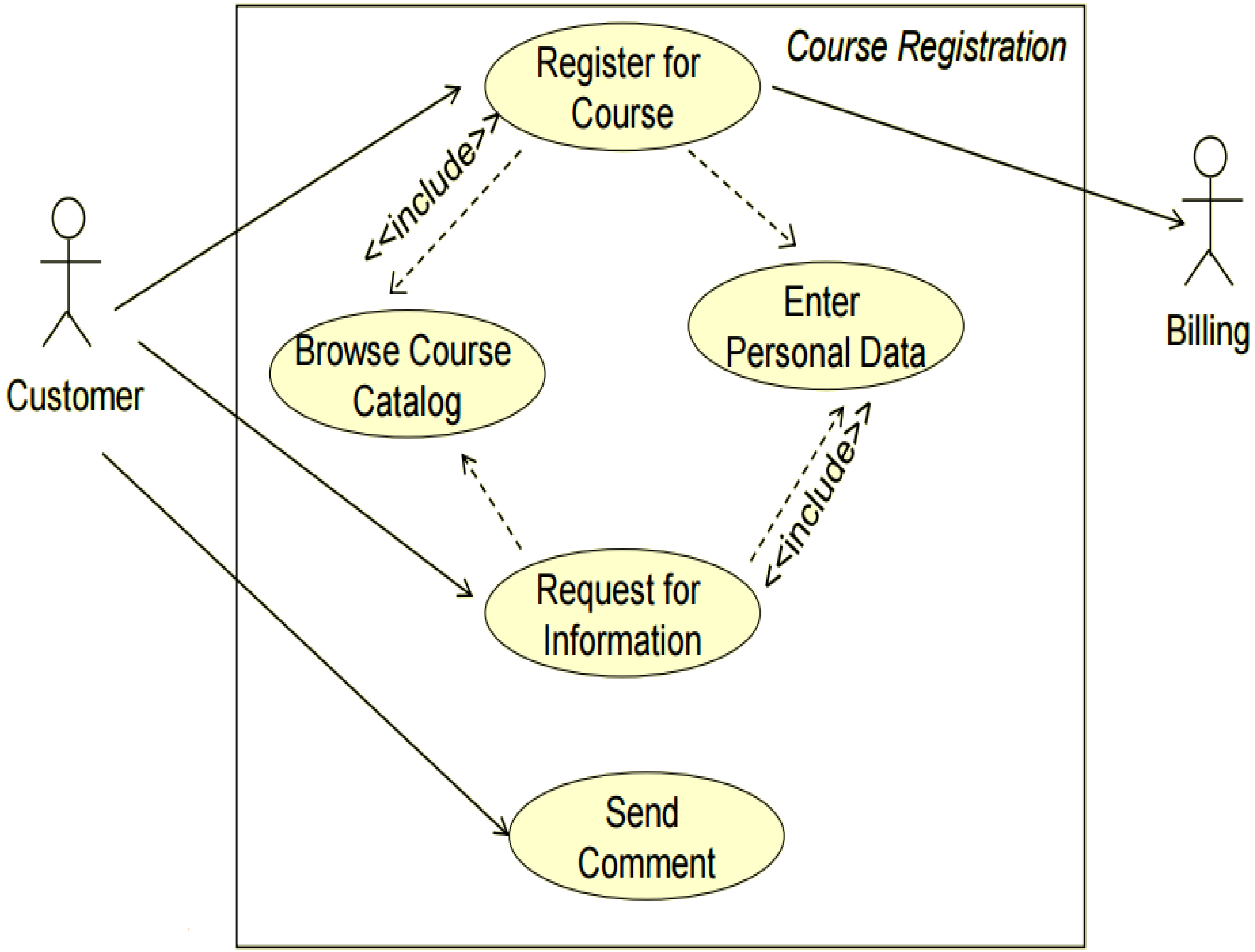
1. Domain requirements may be expressed using special domain terminologies, and software engineers often find it difficult to understand and it’s easy for them to misunderstand.
2. Sometimes stakeholders won’t tell you some requirements because they assume it’s so fundamental and it doesn’t worth mentioning, or they find it difficult to explain, which won’t be taken into consideration in the requirements.

# Use Cases & Scenarios

- The use cases and scenarios are two different techniques, but, usually they are used together.
- Use cases identify interactions between the system and it's users or even other external systems (using graphical notations), while a scenario is a textual description of one or more of these interactions.
- Use case involves some symbols to describe the system:

Use case diagram symbols and an example



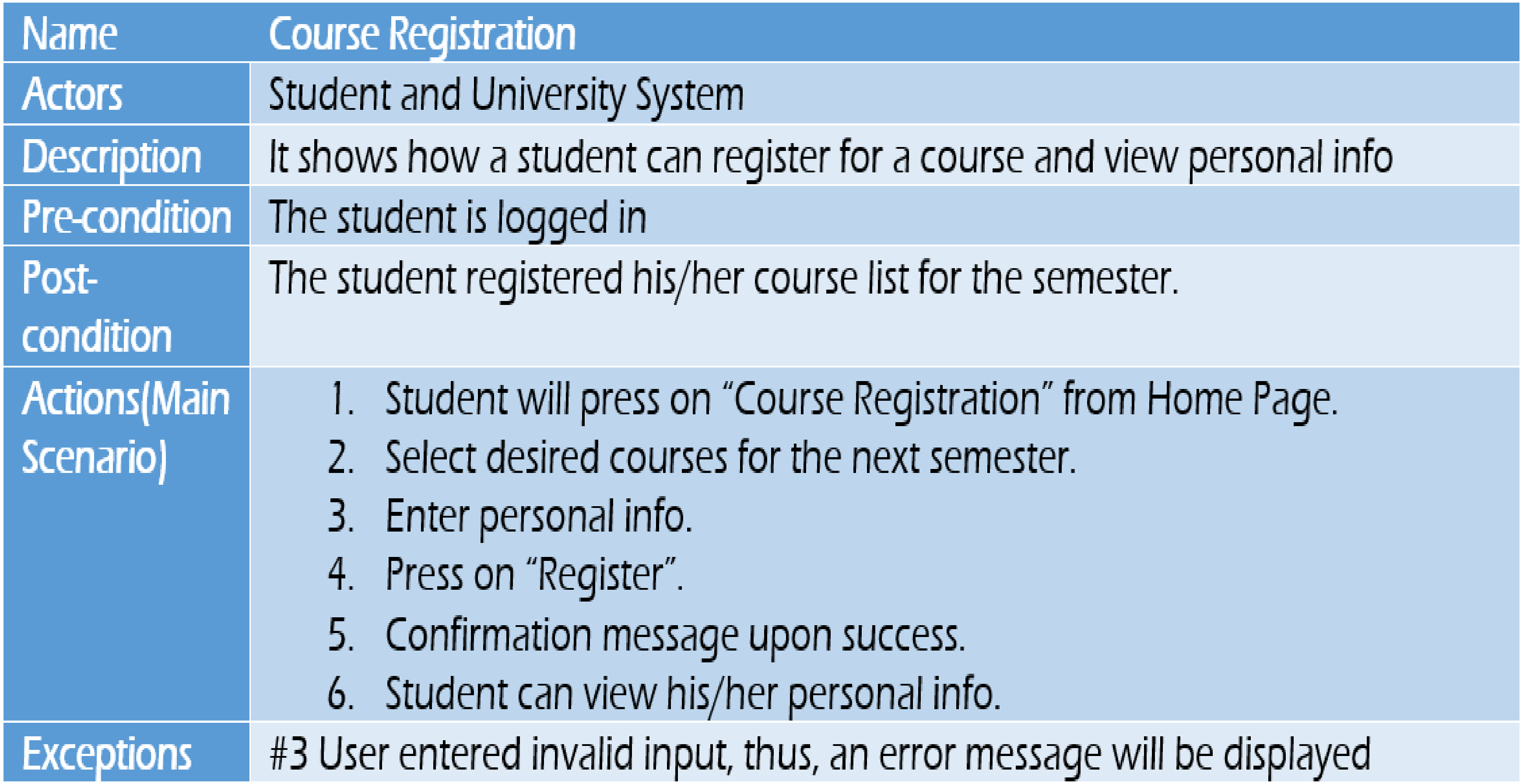

Course Registration
Register for Course
Customer
Billing
<<include>>
Browse Course Catalog
Enter Personal Data
<<include>>
Request for Information
Send Comment

1. **Actors**: Are those who interact with the system; human or other systems
2. **Interaction (Use Case)**: It denotes the name of the interaction (verb). It’s represented as a named ellipse.
3. **Connection**: Lines that links between the actors and the interactions.
4. **Include Relationship**: It denotes a connection between two interactions when an interaction is invoked by another. As an example, splitting a large interaction into several interactions.
5. **Exclude Relationship**: It denotes a connection between two interactions when you want to extend an interaction by adding an optional behavior, but you can use the main interaction on it’s own without the extending interaction.

- Now, we are going to use scenarios to describe the interactions in each use case textually. They should have a format and include the following:

1. A description of the initial situation.
2. A description of the flow of the events or interactions with the system
3. A description of the exceptions, or in other words, what can go wrong, and how it can be handled.
4. Any concurrent activities should be mentioned
5. A description of the final state.

Here is the example for a scenario for the use case example above.



| Name | Course Registration |
|---|---|
| Actors | Student and University System |
| Description | It shows how a student can register for a course and view personal info |
| Pre-condition | The student is logged in |
| Post-condition | The student registered his/her course list for the semester. |
| Actions(Main Scenario) | 1. Student will press on “Course Registration” from Home Page.<br>2. Select desired courses for the next semester.<br>3. Enter personal info.<br>4. Press on “Register”.<br>5. Confirmation message upon success.<br>6. Student can view his/her personal info. |
| Exceptions | #3 User entered invalid input, thus, an error message will be displayed |

- Use case and scenarios are effective techniques for eliciting the requirements. But, because they focus on the interactions with the system, they aren’t effective for eliciting high-level business, non-functional, or domain requirements.
- The next two phases are about analyzing requirements: determining whether the stated requirements are clear, complete, consistent and unambiguous, group related requirements and organize them into related components, and resolving any apparent conflicts.

## Ethnography

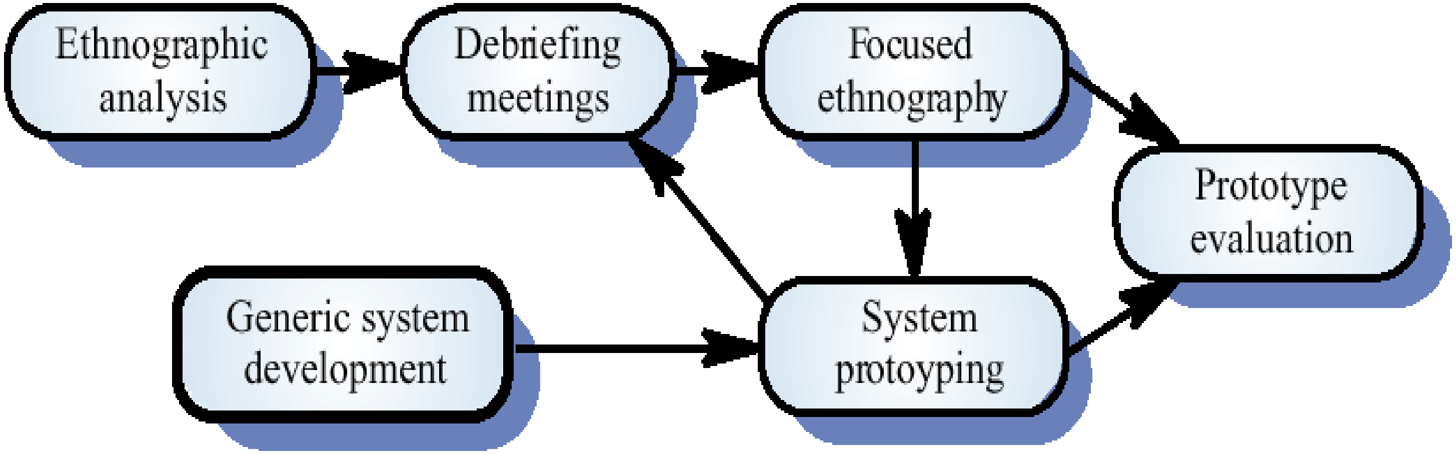
- A social scientists spends a considerable time observing and analysing how people actually work
- People do not have to explain or articulate their work
- Social and organisational factors of importance may be observed
- Ethnographic studies have shown that work is usually richer and more complex than suggested by simple system models

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
- Problem with ethnography is that it studies existing practices which may have some historical basis which is no longer relevant

## Ethnography and prototyping

- 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.
- The problem with ethnography is that it studies existing practices which may have some historical basis which is no longer relevant.



Scope of ethnography:

- Requirements that are derived from the way that people actually work rather than the way I which process definitions suggest that they ought to work
- Requirements that are derived from cooperation and awareness of other people’s activities

## 2. Requirements Classification & Organization

- It's very important to organize the overall structure of the system.
- Putting related requirements together, and decomposing the system into sub components of related requirements. Then, we define the relationship between these components.
- What we do here will help us in the decision of identifying the most suitable architectural design patterns.

## 3. Requirements Prioritization & Negotiation

- We previously explained why eliciting and understanding the requirements is not an easy process.
- One of the reasons is the conflicts that may arise as a result of having different stakeholders involved. *Why?* because it's hard to satisfy all parties, if it's not impossible.

- This activity is concerned with prioritizing requirements and finding and resolving requirements conflicts through negotiations until you reach a situation where some of the stakeholders can compromise.
- We shouldn’t reach a situation where a stakeholder is not satisfied because his requirements is not taken into consideration.
- Prioritizing your requirements will help you later to focus on the essentials and core features of the system, so you can meet the user expectations. It can be achieved by giving every piece of function a priority level. So, functions with higher priorities need higher attention and focus.

**4. Requirements Specification**

- The requirements are then documented.

# REQUIREMENTS VALIDATION

- Concerned with demonstrating that the requirements define the system that the customer really wants
- Requirements error costs are high so validation is very important
  - Fixing a requirements error after delivery may cost up to 100 times the cost of fixing an implementation error

# Requirements checking

- Validity. Does the system provide the functions which best support the customer's needs?
- Consistency. Are there any requirements conflicts?
- Completeness. Are all functions required by the customer included?
- Realism. Can the requirements be implemented given available budget and technology
- Verifiability. Can the requirements be checked?

# Requirements validation techniques

- Requirements reviews
  - Systematic manual analysis of the requirements
- Prototyping
  - Using an executable model of the system to check requirements.
- Test-case generation
  - Developing tests for requirements to check testability
- Automated consistency analysis
  - Checking the consistency of a structured requirements description

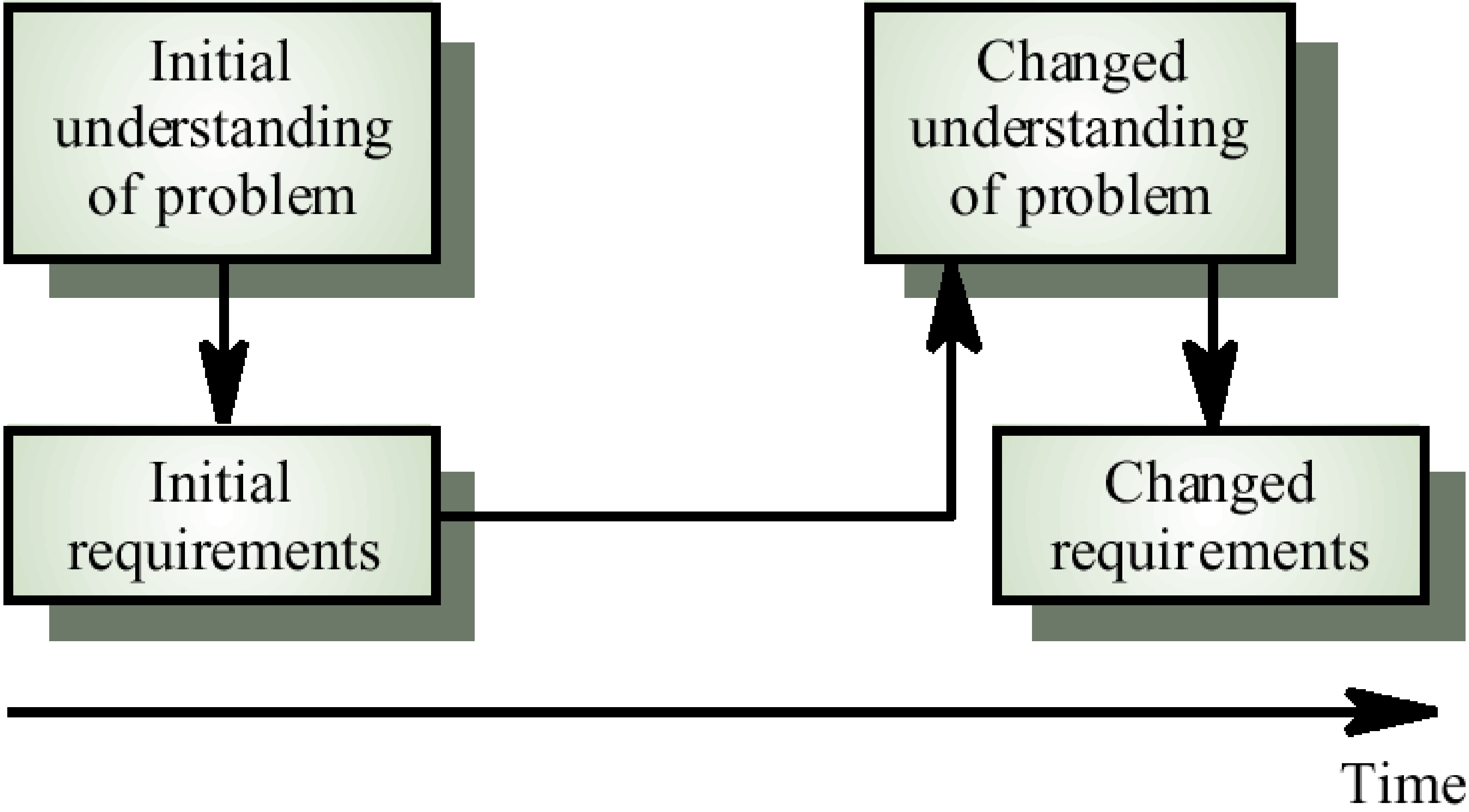
# REQUIREMENTS MANAGEMENT

- Requirements management is the process of managing changing requirements during the requirements engineering process and system development
- Requirements are inevitably incomplete and inconsistent
  - New requirements emerge during the process as business needs change and a better understanding of the system is developed
  - Different viewpoints have different requirements and these are often contradictory

# Requirements change

- The priority of requirements from different viewpoints changes during the development process
- System customers may specify requirements from a business perspective that conflict with end-user requirements
- The business and technical environment of the system changes during its development

# Requirements evolution


Initial understanding of problem
Changed understanding of problem
Initial requirements
Changed requirements
Time

# Requirements management planning

- During the requirements engineering process, you have to plan:
  - Requirements identification
    - How requirements are individually identified
  - A change management process
    - The process followed when analysing a requirements change
  - Traceability policies
    - The amount of information about requirements relationships that is maintained
  - CASE tool support
    - The tool support required to help manage requirements change

# Traceability

- Traceability is concerned with the relationships between requirements, their sources and the system design
- Source traceability
  - Links from requirements to stakeholders who proposed these requirements
- Requirements traceability
  - Links between dependent requirements
- Design traceability
  - Links from the requirements to the design

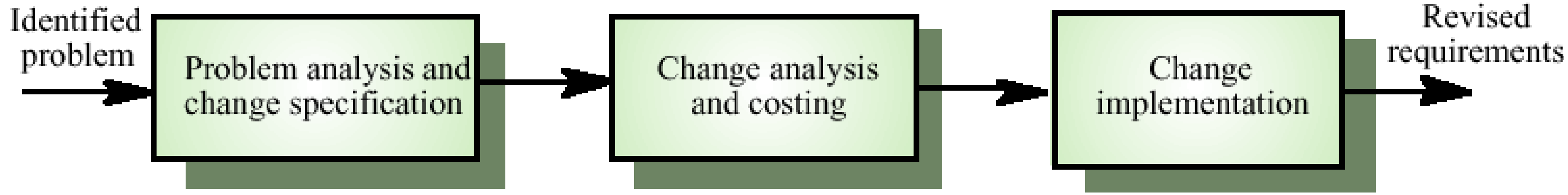
# CASE TOOL SUPPORT

- Requirements storage
  - Requirements should be managed in a secure, managed data store
- Change management
  - The process of change management is a workflow process whose stages can be defined and information flow between these stages partially automated
- Traceability management
  - Automated retrieval of the links between requirements

# Requirements change management

- Should apply to all proposed changes to the requirements
- Principal stages
  - Problem analysis. Discuss requirements problem and propose change
  - Change analysis and costing. Assess effects of change on other requirements
  - Change implementation. Modify requirements document and other documents to reflect change

# Requirements change management


Identified problem
Problem analysis and change specification
Change analysis and costing
Change implementation
Revised requirements

# Key points

- The requirements engineering process includes a feasibility study, requirements elicitation and analysis, requirements specification and requirements management
- Requirements analysis is iterative involving domain understanding, requirements collection, classification, structuring, prioritisation and validation
- Systems have multiple stakeholders with different requirements

# Key points

- Social and organisation factors influence system requirements
- Requirements validation is concerned with checks for validity, consistency, completeness, realism and verifiability
- Business changes inevitably lead to changing requirements
- Requirements management includes planning and change management