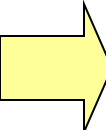# Requirements Engineering Processes

With materials adapted from Software Engineering, 8th Ed. by Ian Sommerville
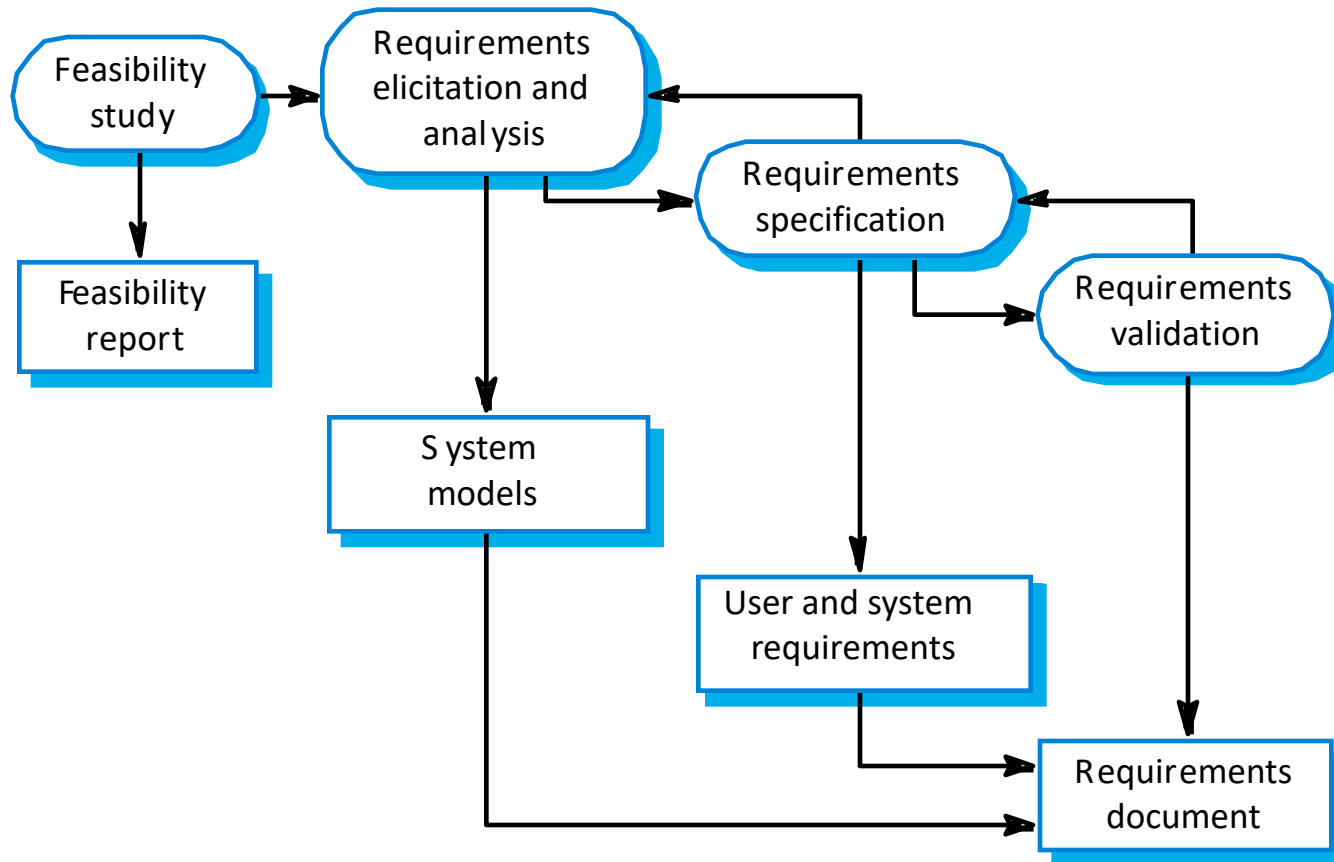
# Requirements Engineering

# Topics covered

- Requirement engineering processes
- Feasibility studies
- Requirements elicitation and analysis
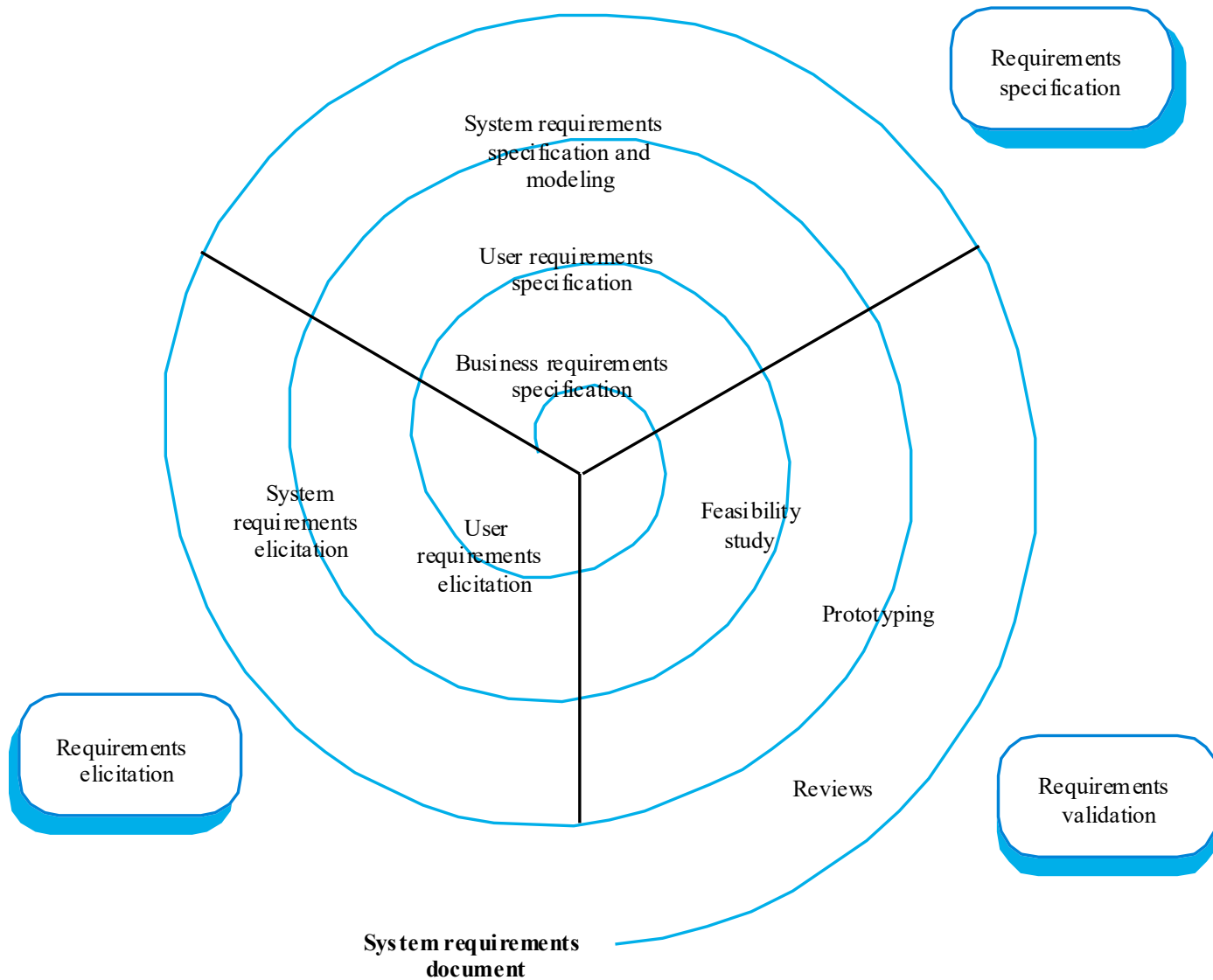- Requirements validation
- Requirements management

# Requirements engineering (RE) processes

- RE process is a set of activities performed to create software requirements
- Processes used for RE vary widely
  - Dependent on organizations, domains, process methodologies

- Common activities
  - Requirements elicitation
  - Requirements analysis
  - Requirements validation
  - Requirements management

# RE process

# Requirements engineering



System requirements specification and modeling

User requirements specification

Business requirements specification

System requirements elicitation

User requirements elicitation

Feasibility study

Prototyping

Reviews

Requirements specification
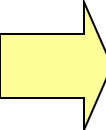
Requirements elicitation

Requirements validation

**System requirements document**

# Topics covered

- Requirement engineering processes
- Feasibility studies
- Requirements elicitation and analysis
- Requirements validation
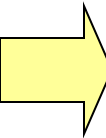- Requirements management

# Feasibility studies

- Feasibility studies decide whether or not the proposed system is worthwhile

- A short focused study checks
    - if the system contributes to organizational objectives
    - if the system can be developed using current technology
    - if the system can be integrated with other systems that are used
    - if the system can be developed within budget and schedule
    - if the system can compete with other similar systems

# Feasibility study implementation

- Based on information assessment (what is required), information collection and report writing

- Questions for people in the organization
  - 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?

- Feasibility report can be included in vision document

# Topics covered

- Requirement engineering processes
- Feasibility studies
- Requirements elicitation and analysis
- Requirements validation
- Requirements management

# Requirements elicitation and analysis

- Also called requirements elicitation or requirements discovery

- Involves technical staff working with customers to find out
  - the application domain
  - the services that the system should provide
  - the system's operational constraints

- May involve end-users, customers, managers, engineers, domain experts, maintainers, operators, etc.
  - also called *stakeholders*

# 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
- Organizational and political factors may influence the system requirements
- Requirements change during the analysis process
  - New stakeholders may emerge and the business environment change

# The requirements spiral
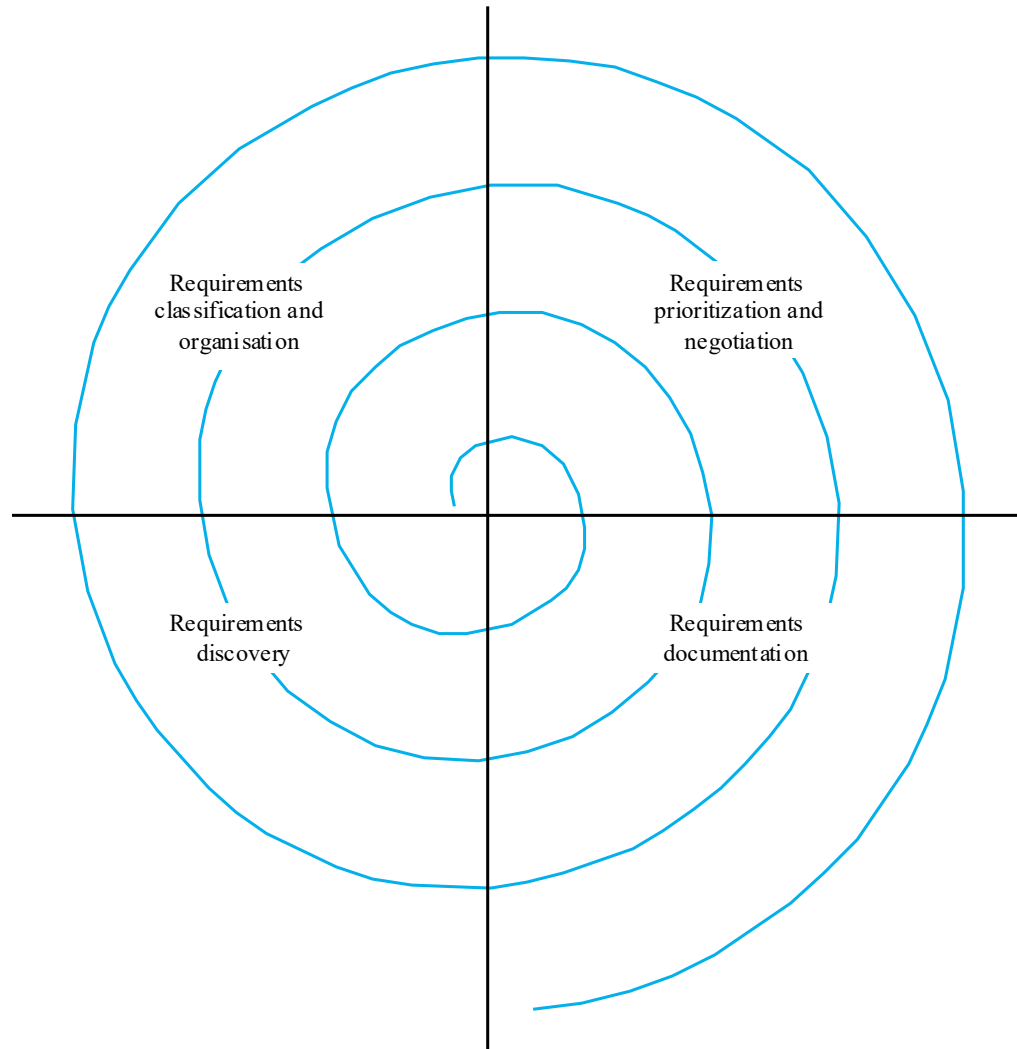
# Process activities

- ## Requirements discovery
  - Interacting with stakeholders to discover their requirements

- ## Requirements classification and organization
  - Groups related requirements and organizes them into coherent clusters

- ## Prioritization and negotiation
  - Prioritizing requirements and resolving requirements conflicts

- ## Requirements documentation
  - Requirements are documented

# Requirements discovery

- Gathering information about the proposed and existing systems
  - extracting user and system requirements from this information

- Sources of information
  - documentation and specifications of similar systems
  - observations
  - existing procedures, processes, and reports
  - other materials

# ATM stakeholders

- Bank customers
- Representatives of other banks
- Bank managers
- Counter staff
- Database administrators
- Security managers
- Marketing department
- Hardware and software maintenance engineers
- Banking regulators

# Viewpoints

- Viewpoints are a way of structuring the requirements to represent perspectives of different stakeholders

- This multi-perspective analysis is important
  - there is no single correct way to analyze system requirements

# Types of viewpoint

- ## Interactor viewpoints
  - People or other systems that interact directly with the system
    - In an ATM, the customer's and the account database are interactor view points

- ## Indirect viewpoints
  - Stakeholders who do not use the system themselves but who influence the requirements
    - E.g., management and security staff are indirect viewpoints

- ## Domain viewpoints
  - Domain characteristics and constraints that influence the requirements
    - E.g., standards for inter-bank communications

# Viewpoint identification

- Identify viewpoints using
  - Providers and receivers of system services
  - Systems that interact directly with the system being specified
  - Regulations and standards
  - Sources of business and non-functional requirements
  - Engineers who have to develop and maintain the system
  - Marketing and other business viewpoints

# LIBSYS viewpoint (VP) hierarchy
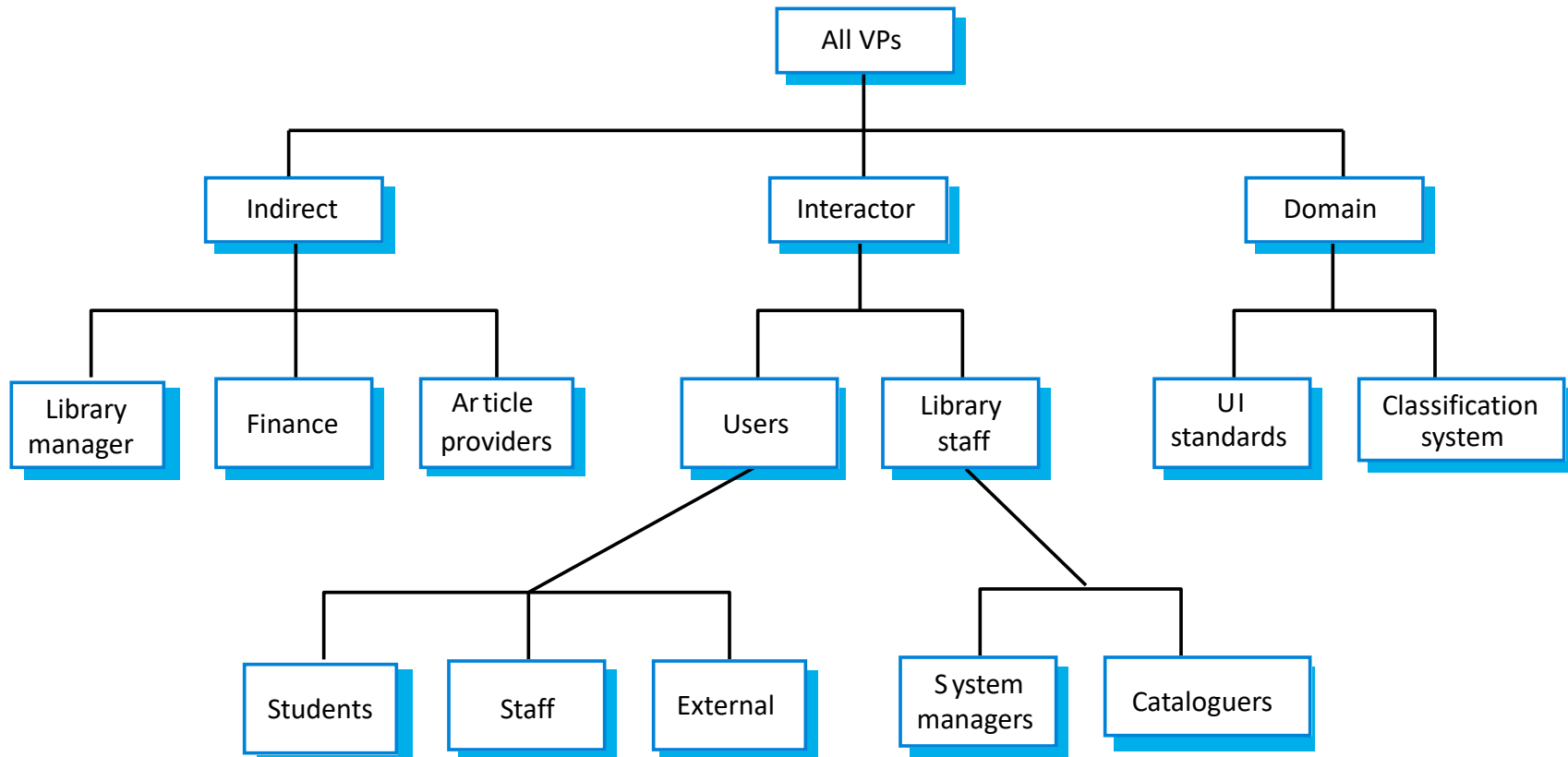
# Interviewing

- RE team questions stakeholders about the existing system and the system to be developed

- Types of interview
    - Closed or structured interviews: prepared questions are answered
    - Open or unstructured interviews: no pre-defined agenda and a range of issues to explored with stakeholders
    - Mixed: some prepared and follow-up questions

# Interviews in practice

- Normally a mix of closed and open-ended interviewing
- Interviews are good for getting an overall understanding of
    - what stakeholders do
    - how they might interact with the system

- Effective interviews
    - Open-minded, willing to listen to stakeholders
    - No pre-conceived ideas about the requirements
    - Prompt interviewees with questions or proposals
    - Avoid asking 'what do you want'

# Use Case Diagrams

- Use case diagrams are used to model the use case view of a system as seen by end users, analysts, and testers

- Use case diagrams are usually a part of requirement specifications

  - Accompanying use case specifications

- Main elements

  - Actors
  - Use cases
  - Relationships

Doctor — Specify Prescription
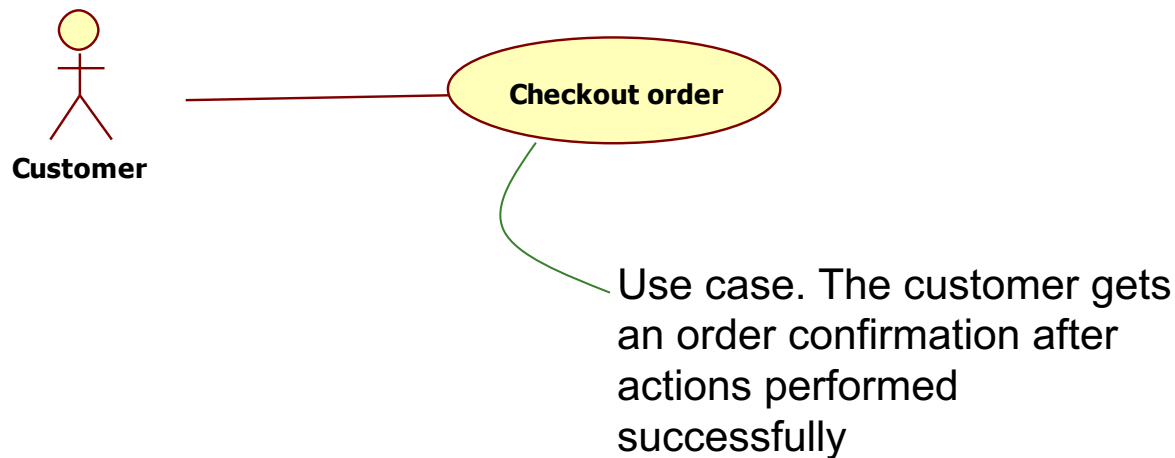
Patient — Provide Health Conditions
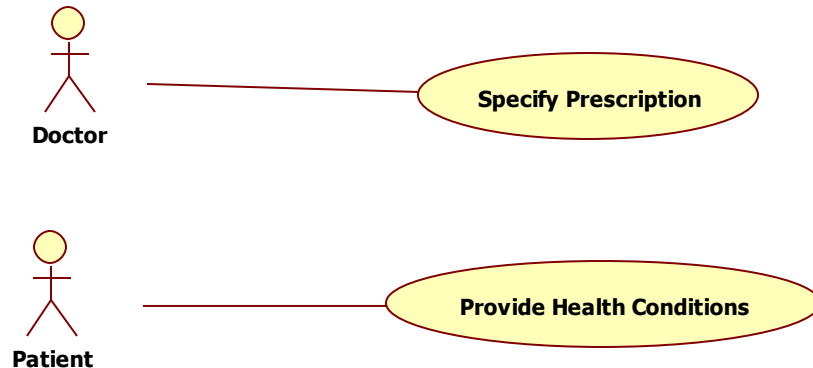
# Actors

- An actor represents an external person, process, or **anything** that interacts with the system

- It represents a role that a person, process, or thing plays with the system

  - One physical user can play multiple roles. Thus, multiple actors can be bound to one physical user
  - Multiple users have the same role, hence, represented by one actor

- Between actors may have a generalization

Employee

Doctor
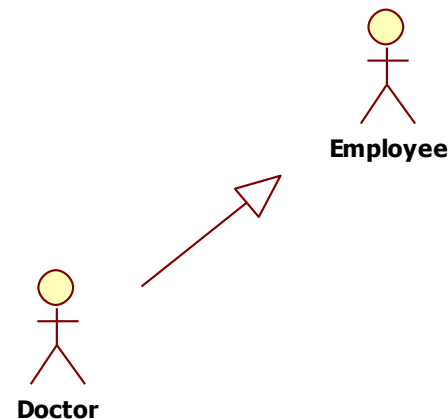
# Use Cases

- A use case describes a set of sequences of actions performed by an actor to produce observable results value to the actor

**Customer**

**Checkout order**

Use case. The customer gets an order confirmation after actions performed successfully
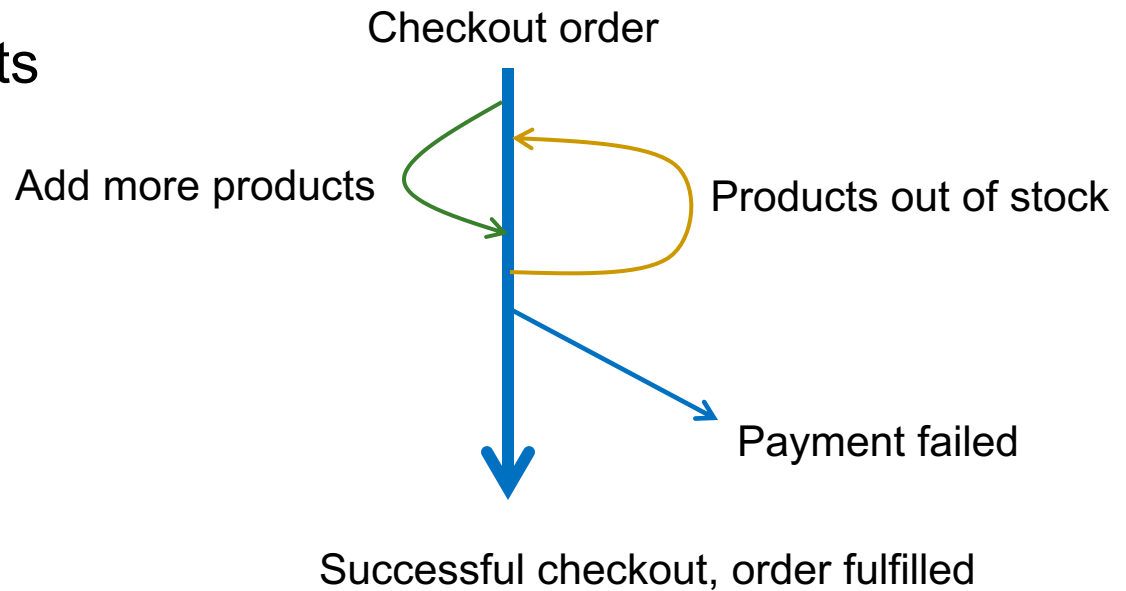
# Use Cases (cont'd)

- At the system level, use cases represent external behavior of a system that is visible to outside users

- A use case specifies WHAT the system does, but not HOW it does

- A use case hides internal structure and operations of a system. It usually represents system-level functions of a system

# Description of Use Cases

- The behavior of a use case is described by a flow of events

- A use case has one main flow (basic flow) and alternative flows
    - Regular variants
    - Odd cases
    - Exceptions

Checkout order

Add more products

Products out of stock

Payment failed

Successful checkout, order fulfilled

# Scenarios

- A use case describes a set of sequences which each sequence in the set represents a possible flow in the use case

- A scenario is a specific sequence of events happening

- A scenario is viewed as an instance of a use case

- Scenarios are real-life examples of how a system can be used

# Scenarios (cont'd)

- Scenarios may 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
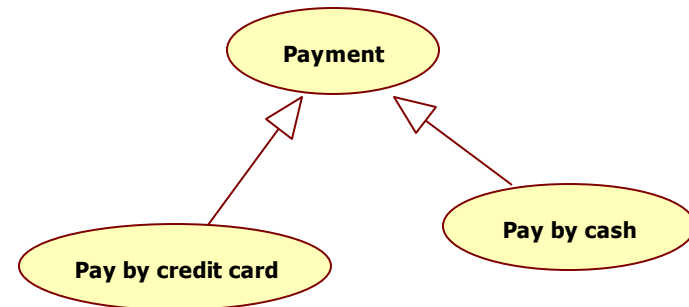
# Types of Use Case Relationships

- ## Association
  - ❑ Between actors and use cases

  **Customer** —— **Checkout order**
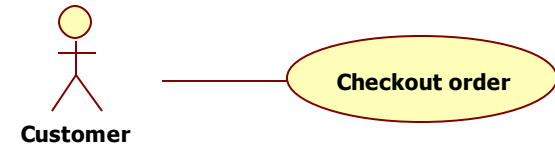
- ## Generalization
  - ❑ Between general use cases and more specific ones

  **Payment**

  **Pay by credit card**     **Pay by cash**

- ## Include
  - ❑ Between use cases
- ## Extend
  - ❑ Between use cases

  **Checkout order** <<include>> **Payment**

  <<extend>>

  **Verify shipping address**

  Difference?
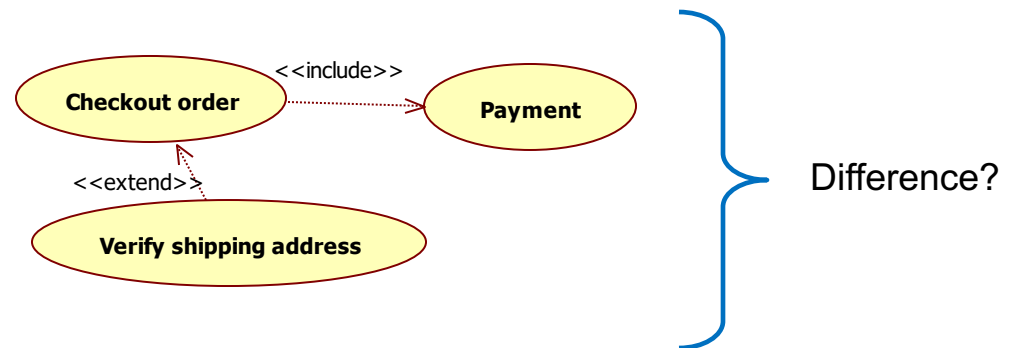
# Exercise – Interview Stakeholders

- One member of Project A plays a role as **customer**
- Everyone else from Project B will interview the **customer** and form a use case model for Project A
  - The customer will answer questions based on his/her knowledge about his/her current project

- You have 15 minutes for interviewing, 5 minutes for reporting (two teams to report)

# Topics covered

- Requirement engineering processes
- Feasibility studies
- Requirements elicitation and analysis
- Requirements validation
- Requirements management
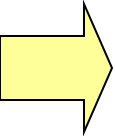
# Requirements validation

- Validate that the requirements reflect what the customer wants

- Errors in requirements result problems in design, code, and test


- Requirements error costs are high
    - Fixing errors caused by incorrect requirements after delivery is much higher than in early stages

# Requirements checking

- Validity
  - Does the system provide the functions which best support the customer's needs?
- Verifiability
  - Can the requirements be checked?
- 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

# Requirements validation techniques

- **Requirements reviews**
  - Systematic manual analysis of the requirements
  - Most common approach

- **Prototyping**
  - Using an executable model of the system to check requirements

- **Test-case generation**
  - Developing tests for requirements to check testability

# Requirements reviews

- Regular reviews should be held while requirements are defined

- Requirements analysts, designers, developers, testers should be involved in reviews

- Reviews may be formal or informal

- Good communications help resolve problems at an early stage

# Topics covered

- Requirement engineering processes
- Feasibility studies
- Requirements elicitation and analysis
- Requirements validation
- Requirements management
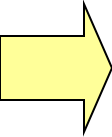
# Requirements management

- ## Requirements management
    - process of managing changing requirements during the project

- ## Requirements are inevitably incomplete and inconsistent
    - New requirements emerge
        - business needs change
        - better understanding of the system

# Requirements change

- New and changed business needs

- Better understanding of the system

- Priority of requirements from different viewpoints changes

- Business and technical environments change during project

# Requirements change management

- **Performed throughout the project**

- **Main steps**
  - Change request
    - Request changes from customers, users, team
  - Change analysis and costing
    - Assess effects of change on other requirements
  - Change implementation
    - Modify requirements and affected docs
    - Change design, code, and test

```
┌─────────────────┐
│  Change         │
│  request        │
└─────────────────┘
         │
         ▼
┌─────────────────┐
│  Change analysis│
│  and costing    │
└─────────────────┘
         │
         ▼
        ◇
   No  ╱ Approved? ╲
 ◄─── ╲           ╱
        ◇
         │ Yes
         ▼
┌─────────────────┐
│  Change         │
│  implementation │
└─────────────────┘
```
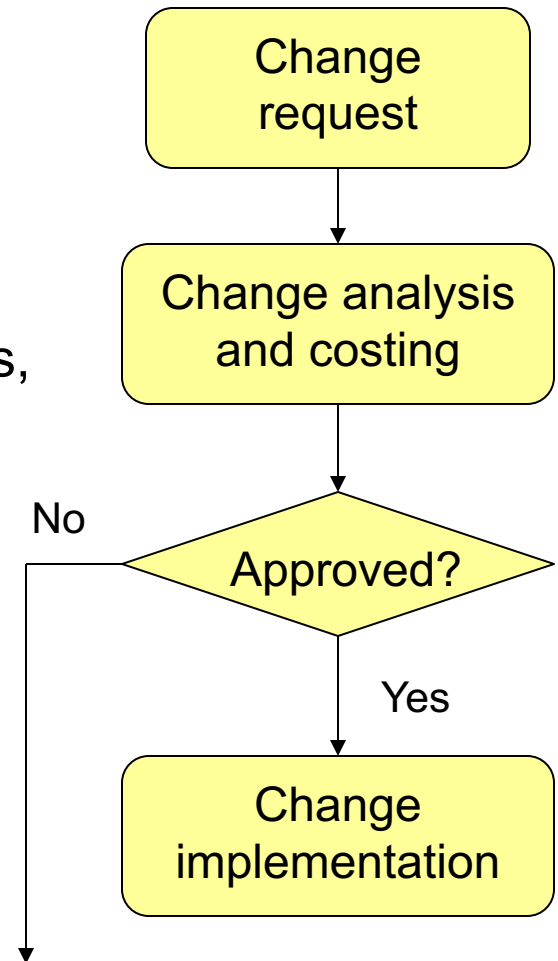
# Key points

- **Requirements engineering process includes**
    - feasibility study, requirements elicitation and analysis, requirements specification and requirements management

- **Requirements elicitation and analysis is iterative involving**
    - domain understanding, requirements collection, classification, structuring, prioritisation and validation.

- **Systems have multiple stakeholders with different requirements**

# Key points

- Social and organization 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